



**TUGAS AKHIR - TE 141599**

## **RANCANG BANGUN PROTOKOL *E-TICKETING***

**PRASETYO YULIANTORO**  
**NRP 2213 106 042**

Dosen Pembimbing  
**Dr. Ir. Endroyono, DEA.**  
**Ir. Gatot Kusrahardjo, MT.**

**JURUSAN TEKNIK ELEKTRO**  
**Fakultas Teknologi Industri**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2015**



**FINAL PROJECT - TE 141599**

## **DESIGN OF E-TICKETING PROTOCOL**

**PRASETYO YULIANTORO**  
**NRP 2213 106 042**

Lecture Advisor  
Dr. Ir. Endroyono, DEA.  
Ir. Gatot Kusrahardjo, MT.

**ELECTRICAL ENGINEERING MAJOR**  
**Industrial Technology Faculty**  
**Sepuluh Nopember Institute of Technology**

**Surabaya 2015**



# **RANCANG BANGUN PROTOCOL *E-TICKETING***

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada  
Bidang Studi Telekomunikasi Multimedia  
Jurusan Teknik Elektro**

**Institut Teknologi Sepuluh Nopember**

**Menyetujui,**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Dr. Ir. Endrovono, DEA**

**NIP. 196504041991021001**

**Ir. Gatot Kusrahardjo, MT.**

**NIP. 195904281986011001**

**SURABAYA,  
JANUARI, 2016**



# RANCANG BANGUN PROTOKOL *E-TICKETING*

**Nama** : Prasetyo Yuliantoro  
**Pembimbing** : Dr. Ir. Endroyono, DEA  
Ir. Gatot Kusrahardjo, MT.

## ABSTRAK

Transportasi modern membutuhkan penerapan Teknologi Informasi dan Komunikasi (TIK), diantaranya adalah teknologi *e-ticketing* yang membutuhkan dukungan perangkat digital dan jaringan komunikasi. Agar pembayaran tiket elektronik dapat berlangsung aman, lancar dan sesuai dengan konsep *management revenue*, maka protokol komunikasi harus dirancang dengan baik. Tugas akhir ini bertujuan merancang protokol untuk sistem *e-ticketing* mulai dari pembacaan data di tiket (RFID), penyimpanan data di *On Board Unit* (OBU), komunikasi dari OBU ke *access point*, hingga komunikasi internet ke *server*.

Perancangan dan implementasi dilakukan dalam 2 tahap, yaitu merancang sistem *e-ticketing* selanjutnya merancang jaringan *e-ticketing*. Perancangan awal membuat simulasi sistem *e-ticketing* pada browser yang terhubung dengan *database*. Simulasi dilakukan dari proses *tap-in*, pengurangan saldo, pengujian keamanan dan monitoring sistem. Selanjutnya dilakukan Perancangan jaringan dengan bantuan software NS-2 yang bertujuan untuk mengetahui *Quality of Service* (QoS) jaringan.

Pengujian sistem *e-ticketing* dilakukan dengan simulasi proses *tap-in* pada *browser* yang didalamnya terdapat *script* yang dapat mengurangi saldo berdasarkan kriteria tarif, dapat memblokir transaksi jika saldo kurang dan dapat menampilkan peta posisi kendaraan. Pada pengujian jaringan *e-ticketing*, didapat hasil paling besar dari throughput sebesar 715,832 kbps, *delay* sebesar 1,24432 ms, *Jitter* sebesar 583,730 ms. Packet loss sebesar 3,8628 %. Berdasarkan hasil QoS tersebut, terdapat satu nilai yang kurang bagus tetapi secara keseluruhan protokol yang dirancang sudah memenuhi kualitas yang baik

**Kata kunci** : E-ticketing, NS-2, Xampp, QoS

## ***DESIGN OF E-TICKETING PROTOCOL***

**Name** : Prasetyo Yuliantoro  
**Supervisors** : Dr. Ir. Endroyono, DEA  
Ir. Gatot Kusrahardjo, MT.

### ***ABSTRACT***

*Modern transportation needs to use information and communication technology including e-ticketing which require support from digital device and communication network. In order this electronic ticketing payment running securely, fluent and same as management revenue concept, hence the communication protocol should be designed well. The purpose of this final project is designing e-ticketing protocol system form reading data with barcode, saving data in On Board Unit, communication from OBU to access point until communication from internet to server.*

*Design and implementation will be done in two steps, the first step is design e-ticketing system and then design e-ticketing network. The initial design is creating e-ticketing system simulation on browser that can connected to database server. This simulation conducted from tap-in process, reducing balance, security testing and system monitoring. After that, beginning network design with NS-2 software simulator which purpose to know the quality of service.*

*The testing will be done by simulation in browser who that script can do tap-in process that the output can be reducing ticket balance based on price regulations, can reject transaction if ticket balance is insufficient and can be display map with position of the vehicle. On testing e-ticketing network, the result of throughput is 715,832 kbps, delay is 1,24432 ms, jitter is 583,730 ms and packet loss is 3,8628 %. From the QoS result, that is one value that is less good, but for overall this protocols has good quality.*

**Keywords** : E-Ticketing, NS2, Xampp, QoS

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat, petunjuk, pengetahuan, serta karunia-Nya sehingga tugas akhir ini dapat terselesaikan. Tugas akhir ini disusun untuk memenuhi syarat menyelesaikan pendidikan Strata-1 pada Bidang Studi Telekomunikasi Multimedia, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya. Judul tugas akhir ini adalah :

### **“RANCANG BANGUN PROTOKOL *E-TICKETING*”**

Terselesaikannya tugas akhir ini tentunya tak lepas dari dorongan dan uluran tangan berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih kepada beberapa pihak yang memberikan dukungan selama proses penyelesaian tugas akhir ini, antara lain :

1. Kedua orang tua penulis tersayang, Ibu Nuraeni Dwi Astuti dan Bapak Wakidi yang telah mensupport banyak hal, serta Reza Edwien Sulistyaningtyas yang telah memberi saran, menghibur dan mendoakan.
2. Bapak Dr. Ir. Endroyono, DEA serta Bapak Ir. Gatot Kusrahardjo, MT. selaku Dosen Pembimbing yang telah banyak memberikan ilmu, pengarahan dan bimbingan selama penyelesaian tugas akhir ini.
3. Teman – teman Lintas Jalur TMM-ITS angkatan 2013 Joko GG, Depa, Tiyan, Bambang, Danar, Banyu, Nita, Tania, Dita, Umcu, Dessy, Mba Dwi dan Sherly.

Semoga tugas akhir ini dapat bermanfaat dan berguna bagi pembacanya. Penulis menyadari dalam penyusunan tugas akhir ini masih jauh dari sempurna sehingga saran, kritik dan diskusi untuk pengembangan dari tugas akhir ini sangat penulis harapkan.

Surabaya, Januari 2016

Penulis

## DAFTAR ISI

ABSTRAK.....	iii
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	1
1.3 Batasan Masalah .....	1
1.4 Tujuan.....	2
1.5 Metodologi.....	2
1.6 Sistematika Penulisan .....	3
1.7 Manfaat.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Angkutan Massal Cepat (AMC) .....	5
2.2 Konsep <i>E-ticketing</i> .....	6
2.3 Teknologi elektronik dan <i>smart-card</i> .....	9
2.5.1 Teknologi <i>Smart-Card</i> .....	9
2.5.2 <i>Memory</i> .....	11
2.4 Protokol Komunikasi E-Ticketing.....	11
2.4.1 <i>Data-link Layer</i> .....	11
2.4.2 <i>Transport Layer</i> .....	12
2.4.3 <i>Application Layer</i> .....	12
2.5 Quality of Service .....	13
2.5.1 <i>Throughput</i> .....	13
2.5.2 Packet Loss.....	14
2.5.3 <i>Delay</i> .....	14
2.5.4 <i>Jitter</i> .....	15
2.6 Xampp .....	16
2.6.1 Apache .....	16
2.6.2 PHP .....	17
2.6.3 MySQL.....	17
2.6.4 PHPMyAdmin.....	17
2.7 NS2.....	17
2.8 Sistem <i>E-Ticketing</i> di Berbagai Kota Besar di Dunia .....	19
2.8.1 London .....	19

2.8.2	Tokyo, Jepang.....	19
2.8.3	Singapura .....	20
2.9	<i>Security</i> Tiket Elektronik.....	21
BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM.....		23
3.1	Perancangan Sistem <i>E-Ticketing</i> .....	23
3.1.1	Proses Komunikasi Tiket – OBU.....	23
3.1.2	Proses Komunikasi OBU – Halte.....	25
3.1.3	Proses Komunikasi Halte – Server.....	27
3.2	AMC Kota Surabaya .....	27
3.3	Sistem Pentarifan <i>E-Ticketing</i> .....	28
3.4	Implementasi Sistem <i>E-ticketing</i> .....	29
3.4.1	Implementasi Sistem.....	29
3.4.2	Implementasi Jaringan .....	35
3.5	Skenario Pengujian Sistem E-ticketing .....	36
3.5.1	Skenario Pengujian Sistem .....	36
3.5.2	Skenario Pengujian Jaringan.....	37
BAB IV PENGUJIAN DAN ANALISIS .....		39
4.1	Pengujian Sistem <i>E-ticketing</i> .....	39
4.1.1	Pengujian Fungsional Sistem.....	39
4.1.2	Pengujian Keamanan Sistem.....	41
4.1.3	Pengujian Monitoring Sistem .....	42
4.1.4	Analisis Hasil Pengujian .....	44
4.2	Pengujian Jaringan <i>E-ticketing</i> .....	45
4.2.1	Pengujian dan analisis <i>throughput</i> jaringan .....	45
4.2.2	Pengujian dan Analisis <i>packet loss</i> jaringan .....	48
4.2.3	Pengujian dan Analisis <i>delay</i> jaringan .....	49
4.2.4	Pengujian dan Analisis <i>jitter</i> jaringan.....	52
BAB V PENUTUP .....		55
5.1	Kesimpulan.....	55
5.2	Saran.....	56
DAFTAR PUSTAKA .....		57
LAMPIRAN .....		59
RIWAYAT HIDUP .....		115



## TABLE OF CONTENT

ABSTRAK.....	iii
<i>ABSTRACT</i> .....	v
PREFACE .....	vii
TABLE OF CONTENT .....	ix
ILLUSTRATION.....	xi
TABLES .....	xiii
CHAPTER 1 PRELIMINARY .....	1
1.1 Background .....	1
1.2 Problems .....	1
1.3 Limitations .....	1
1.4 Purpose .....	2
1.5 Methods .....	2
1.6 Systemathical Study.....	3
1.7 Relevance.....	3
CHAPTER II TEORY .....	5
2.1 Mass Rapid Transport.....	5
2.2 E-ticketing concept .....	6
2.3 Smart-card.....	9
2.5.1 Smart-card Technology .....	9
2.5.2 Memory .....	11
2.4 Protocol of E-Ticketing .....	11
2.4.1 Data-link Layer .....	11
2.4.2 Transport Layer .....	12
2.4.3 Application Layer.....	12
2.5 Quality of Service .....	13
2.5.1 Throughput.....	13
2.5.2 Packet Loss.....	14
2.5.3 Delay .....	14
2.5.4 Jitter.....	15
2.6 Xampp .....	16
2.6.1 Apache .....	16
2.6.2 PHP .....	17
2.6.3 MySQL.....	17
2.6.4 PHPMyAdmin.....	17
2.7 NS2 .....	17
2.8 E-ticketing System in Many Big Cities in The World .....	19

2.8.1	London .....	19
2.8.2	Tokyo, Jepang .....	19
2.8.3	Singapura .....	20
2.9	Security E-ticketing .....	21
CHAPTER III DESAIN AND SYSTEM IMPLEMENTATION .....		23
3.1	Desain System E-Ticketing .....	23
3.1.1	Communication Between Tiket – OBU .....	23
3.1.2	Communication Between OBU – Halte .....	25
3.1.3	Communication Between Halte – Server .....	27
3.2	MRT in Surabaya City .....	27
3.3	E-Ticketing Pricing System .....	28
3.4	E-ticketing Implementation System .....	29
3.4.1	System Implementation .....	29
3.4.2	Network Implementation .....	35
3.5	Test Scenario E-ticketing System .....	36
3.5.1	System Test Scenario .....	36
3.5.2	Network Test Scenario .....	37
CHAPTER IV TESTING AND ANALYSIS .....		39
4.1	E-ticketing System Test .....	39
4.1.1	Functional System Test .....	39
4.1.2	Security System Test .....	41
4.1.3	Monitoring System Test .....	42
4.1.4	Analysis .....	44
4.2	E-ticketing Network Test .....	45
4.2.1	Test and Analysis of Network Throughput .....	45
4.2.2	Test and Analysis of Network Packet Loss .....	48
4.2.3	Test and Analysis of Network Delay .....	49
4.2.4	Test and Analysis of Network Jitter .....	52
CHAPTER V CLOSING .....		55
5.1	Conclusion .....	55
5.2	Suggestion .....	56
BIBLIOGRAPHY .....		57
ATTACHMENT .....		59
BIOGRAPYH .....		115

## DAFTAR GAMBAR

<b>Gambar 3.1</b> Skema e-ticketing secara umum .....	23
<b>Gambar 3.2</b> Komunikasi dari tiket ke OBU .....	24
<b>Gambar 3.3</b> Flowchart proses tap-in.....	25
<b>Gambar 3.4</b> Komunikasi dari OBU ke Access Point.....	25
<b>Gambar 3.5</b> Flowchart komunikasi dari OBU ke Halte.....	26
<b>Gambar 3.6</b> Jalur Monorail dan Tram .....	27
<b>Gambar 3.7</b> Flowchart pembuatan sistem e-ticketing.....	30
<b>Gambar 3.8</b> Daftar database .....	31
<b>Gambar 3.9</b> Tampilan tabel kendaraan .....	31
<b>Gambar 3.10</b> Tampilan tabel tiket .....	32
<b>Gambar 3.11</b> Tampilan tabel rekap .....	32
<b>Gambar 3.12</b> Tampilan tabel halte.....	33
<b>Gambar 3.13</b> Flowchart simulasi sistem e-ticketing.....	34
<b>Gambar 3.14</b> Flowchart perancangan protokol e-ticketing.....	35
<b>Gambar 3.15</b> Hasil rancangan node yang sudah jadi .....	36
<b>Gambar 4.1</b> Tampilan halaman ‘Update Live’ .....	39
<b>Gambar 4.2</b> Tampilan bagian tap-in .....	40
<b>Gambar 4.3</b> Tampilan hasil tap-in .....	40
<b>Gambar 4.4</b> Tampilan halaman ‘Update Live’ setelah refresh .....	40
<b>Gambar 4.5</b> Tampilan bagian tap-out. ....	41
<b>Gambar 4.6</b> Tampilan halaman ‘Update Live’ setelah proses tap-out. ....	41
<b>Gambar 4.7</b> Tiket 1002 dengan saldo sebesar 2000 .....	41
<b>Gambar 4.8</b> Alert sistem .....	42
<b>Gambar 4.9</b> Posisi kendaraan pada peta Kota Surabaya.....	44
<b>Gambar 4.10</b> Grafik polynomial troughput terhadap jarak pada jalur monorail pengujian tahap pertama .....	46
<b>Gambar 4.11</b> Grafik polynomial troughput terhadap jarak pada jalur tram pengujian tahap pertama .....	46
<b>Gambar 4.12</b> Grafik perbandingan troughput terhadap jarak pada jalur monorail pengujian tahap kedua .....	47
<b>Gambar 4.13</b> Grafik perbandingan troughput terhadap jarak pada jalur tram pengujian tahap kedua .....	47
<b>Gambar 4.14</b> Grafik perbandingan packet loss terhadap jarak pada jalur monorail .....	48
<b>Gambar 4.15</b> Grafik perbandingan packet loss terhadap jarak pada jalur tram .....	49



**Gambar 4.16** Grafik polynomial delay terhadap jarak pada jalur monorail pengujian tahap pertama .....50

**Gambar 4.17** Grafik polynomial delay terhadap jarak pada jalur tram pengujian tahap pertama .....50

**Gambar 4.18** Grafik perbandingan delay terhadap jarak pada jalur monorail pengujian tahap kedua .....51

**Gambar 4.19** Grafik perbandingan delay terhadap jarak pada jalur tram pengujian tahap kedua .....52

**Gambar 4.20** Grafik polynomial jitter terhadap jarak pada jalur monorail pengujian tahap pertama .....52

**Gambar 4.21** Grafik polynomial jitter terhadap jarak pada jalur tram pengujian tahap pertama .....53

**Gambar 4.22** Grafik perbandingan jitter terhadap jarak pada jalur monorail pengujian tahap kedua .....54

**Gambar 4.23** Grafik perbandingan jitter terhadap jarak pada jalur tram pengujian tahap kedua .....54

## DAFTAR TABEL

<b>Tabel 2.1</b> Lingkup sistem e-ticketing .....	7
<b>Tabel 2.2</b> Standar packet loss versi TIPHON .....	14
<b>Tabel 2.3</b> Standar packet loss versi ITU-T .....	14
<b>Tabel 2.4</b> Standar delay versi TIPHON .....	15
<b>Tabel 2.5</b> Standar delay versi ITU-T .....	15
<b>Tabel 2.6</b> Standar jitter versi TIPHON .....	16
<b>Tabel 3.1</b> Klasifikasi pentarifan e-ticketing .....	28
<b>Tabel 4.1</b> Halte dan koordinat halte.....	42
<b>Tabel 4.2</b> Hasil simulasi .....	44

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Transportasi modern saat ini, banyak membutuhkan aplikasi Teknologi Informasi dan Komunikasi (TIK), diantaranya adalah aplikasi mengenai tiket elektronik yang membutuhkan dukungan perangkat digital dan jaringan komunikasi. Komunikasi dari halte ke *server* membutuhkan layanan data yang baik, lancar dan cepat.

*Electronic ticketing* dirancang di kota Surabaya pada *Surabaya Mass Rapid Transportation (SMART) Tram and Monorail Project* dengan jumlah halte sebanyak 50 halte. Semua halte terhubung dengan *server/sentral*. Komunikasi dari halte ke sentral berlangsung dari data yang terdapat di kartu/tiket yang berbasis *smartcard* dibaca oleh perangkat OBU (*On Board Unit*), dari OBU data akan diteruskan ke *access point* yang berada di halte dan dari halte diteruskan ke *server/sentral*.

Untuk menghubungkan semua halte tersebut diperlukan adanya topologi jaringan dan sistem yang sesuai. Sistem tersebut meliputi sistem *e-ticketing*, dimana sistem tersebut dapat melakukan proses rekapitulasi untuk semua tiket.

Oleh karena itu, diperlukan protokol komunikasi yang mampu memenuhi kebutuhan transportasi modern. Dengan dibutuhkannya layanan yang seperti itu, maka dibuatlah protokol komunikasi dari kendaraan ke *server*.

### **1.2 Perumusan Masalah**

Masalah yang akan dibahas dalam Tugas Akhir ini adalah :

1. Merancang sistem *e-ticketing* dan jaringan *e-ticketing*
2. Mengukur kehandalan sistem *e-ticketing* dan QoS jaringan meliputi *throughput*, *packet loss*, *delay* dan *jitter*.

### **1.3 Batasan Masalah**

Permasalahan dibatasi sebagai berikut :

1. Studi kasus untuk perancangan transportasi dengan asumsi dilakukan di Kota Surabaya.
2. Simulasi *e-ticketing* dilakukan menggunakan *server localhost*
3. *Software* yang digunakan NS-2 dan XAMPP.



## 1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah melakukan perancangan protokol *e-ticketing* pada Kota Surabaya sehingga dapat menghasilkan rancangan dengan kualitas layanan yang baik

## 1.5 Metodologi

Dalam proses pengerjaan penelitian Tugas Akhir ini dapat dilakukan dengan mengelompokkan dalam beberapa metodologi, yaitu :

### 1. Studi Literatur

Pencarian dan pengumpulan literatur yang berkaitan dengan perancangan protokol sistem *e-ticketing*, baik berupa jurnal artikel, buku referensi, internet dan dari sumber yang lain.

### 2. Perancangan dan implementasi protokol *E-ticketing*

Pada tahap ini, dilakukan perancangan protokol komunikasi *e-ticketing*. Perancangan tersebut berupa perancangan sistem *e-ticketing* dan perancangan jaringan komunikasi *e-ticketing* di Kota Surabaya. Setelah perancangan selesai, kemudian dilanjutkan dengan implementasi pada masing-masing software.

### 3. Implementasi Sistem *E-ticketing*

Sistem *e-ticketing* dirancang menggunakan *software* Xampp dalam bentuk *website*. *Database* dan *script* yang sudah dirancang, dicoba untuk diimplementasikan dalam proses *tap-in* sampai dengan proses *tap-out*. Implementasi tersebut dilakukan pada *browser*.

### 4. Implementasi Jaringan *E-ticketing*

Implementasi jaringan *e-ticketing* mengikuti desain AMC Kota Surabaya. Pada desain tersebut terdapat 50 titik *node* yang digambarkan dan disimulasikan pada *software* NS2. Setelah implementasi, selanjutnya dilakukan pengujian QoS

### 5. Analisis Hasil

Hasil dari kedua implementasi selanjutnya dianalisis. Pada implementasi sistem *e-ticketing* dianalisis tentang fungsi nya, sedangkan pada jaringan *e-ticketing* dianalisis tentang QoS meliputi *throughput*, *delay*, *packet loss* dan *jitter*.

## 1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini adalah sebagai berikut:

### BAB I PENDAHULUAN

Pada bab ini akan diuraikan latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi dan manfaat

### BAB II TINJAUAN PUSTAKA

Pada bab ini akan dibahas landasan teori mengenai konsep angkutan masal cepat, konsep *e-ticketing*, protokol komunikasi *e-ticketing*, QoS (*Quality of Service*) jaringan, konsep XAMPP dan NS2, sistem *e-ticketing* di berbagai negara dan juga *security* tiket elektronik

### BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan tentang perancangan dan implementasi sistem *e-ticketing*. Sistem *e-ticketing* di rancang dan di implementasikan menggunakan *software* Xampp dan jaringan komunikasi *e-ticketing* dirancang dan di implementasikan menggunakan *software network simulator 2* (NS2)

### BAB IV PENGUJIAN DAN ANALISIS

Pada bab ini berisi hasil dari rancangan yang sudah di buat pada bab III. Dari pengujian ini kemudian dianalisis dan ditarik kesimpulan sementara mengenai parameter-parameter yang telah diuji

### BAB V PENUTUP

Pada bab ini berisi kesimpulan dari keseluruhan materi dan dari hasil analisis data pada bab IV. Selain itu pada bab ini dibahas mengenai saran yang bisa dilakukan untuk pengembangan penelitian selanjutnya

## 1.7 Manfaat

Hasil yang diperoleh dari tugas akhir ini diharapkan dapat memberi manfaat, yakni :

1. Memberikan rancangan mengenai konsep *e-ticketing*
2. Memberikan gambaran mengenai sistem transaksi *e-ticketing* beserta *database*-nya
3. Memberikan gambaran mengenai perancangan jaringan pada komunikasi dari halte hingga *server*

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Angkutan Massal Cepat (AMC)**

Angkutan umum memiliki peran yang sangat penting bagi penduduk suatu kota untuk bisa secara efektif memberikan akses bagi barang dan jasa. Sistem angkutan ini diperlukan bagi kota-kota besar karena permasalahan transportasi yang tidak tertata secara bagus, dapat menambah angka kemacetan lalu-lintas dan buruknya pelayanan angkutan umum. Pada negara maju, angkutan ini terdiri dari beberapa moda transportasi, diantaranya adalah moda transportasi berupa kereta cepat, bus yang dapat menampung penumpang dalam jumlah yang banyak, dan lain-lain. Beberapa tujuan diterapkannya angkutan massal cepat ini adalah

- Meningkatkan kualitas layanan dan keselamatan lalu-lintas
- Menurunkan tingkat kemacetan
- Meningkatkan efisiensi transportasi
- Mengurangi polusi udara
- Meningkatkan efisiensi pemanfaatan energi

Sistem angkutan massal yang umum dipergunakan di perkotaan adalah sebagai berikut :

1. *Bus Rapid Transit (BRT)* - Teknologi berbasis Bis, pada umumnya beroperasi pada jalur khusus yang sebidang dengan permukaan jalan yang ada, pada kondisi tertentu (mis persimpangan atau pusat kota) yang diperlukan pemisahan elevasi, BRT dilewatkan terowongan atau jembatan khusus.
2. *Light Rail Transit (LRT)* - Teknologi berbasis Rel-Listrik, pada umumnya beroperasi menggunakan kendaraan rel tunggal atau kereta listrik pendek di jalur rel khusus sebidang dengan permukaan tanah dengan konektor listrik di atas kendaraan. Jenis lain dari LRT adalah Tram System, pada umumnya dengan ukuran kendaraan yang lebih kecil dan beroperasi di jalur jalan raya tanpa pemisahannya dengan lalu-lintas lainnya.
3. *Underground Metro* - Teknologi berbasis kereta api (heavy rail) beroperasi pada jalur di bawah permukaan tanah atau terowongan.
4. *Elevated Rail Transit* - Teknologi berbasis kereta api (heavy rail) beroperasi pada jalur di atas permukaan tanah atau jalan layang.



5. *Suburban Rail* - Teknologi berbasis kereta api yang beroperasi pada jalur khusus di permukaan tanah atau di atas permukaan tanah, pada umumnya melayani penumpang dari pinggiran kota ke kota.
6. *Personal Rapid Transit (PRT)* - Teknologi berbasis rel atau roda, mengangkut penumpang dengan kendaraan berfasilitas AVG (*automatic guided vehicles*) yang beroperasi pada jalur khusus.

## 2.2 Konsep *E-ticketing*

Beberapa konsep dari teknologi *e-ticketing* diklarifikasikan menurut cara mereka digunakan untuk pembayaran. Lebih dekat kartu dengan sistem pembayaran, lebih dapat diandalkan untuk bertransaksi, tetapi membatasi gerak untuk pengguna. Dalam konsep ini, *e-ticketing* dapat dibagi menjadi seperti berikut ini

- *Contact-based technology* didasarkan pada komunikasi standart antara perangkat pengguna (hanya memori atau smart card) dan sistem akses sesuai dengan standart ISO 7816
- Teknologi *proximity* seringkali didasarkan pada komunikasi *contactless* sesuai dengan *sub-standart* yang berbeda dari ISO 14443, yang menghasilkan jarak transmisi sekitar 10cm
- Teknologi sekitar (*Vicinity*) terkait dengan ISO 15693 dan biasanya mencakup jarak transmisi hingga 1m
- *Long-range*, teknologi memerlukan baterai di perangkat pengguna (kartu) dan menggabungkan kopling induktif dengan transmisi data frekuensi radio sedangkan metode komunikasi pertama digunakan untuk mengaktifkan perangkat pengguna ketika memasuki kendaraan transportasi, yang kedua memungkinkan transmisi data *contactless* antara semua tempat di dalam kendaraan dan, misalnya, komponen akses elektronik di langit-langit. Teknologi ini menyediakan mekanisme anti-tabrakan untuk mencegah tabrakan transaksi elektronik, karena dapat terjadi sebaliknya

Konsep yang lain yaitu tentang sudut pandang penumpang. Beberapa penumpang memiliki tarif tersendiri atau diskon, seperti:

- Anak-anak
- Pelajar
- Orang tua dan pensiunan
- Penyandang cacat
- Polisi dan tentara

**Tabel 2.1** Lingkup sistem *e-ticketing*

lingkup sistem <i>e-ticketing</i>	
<i>Open payment schemes</i>	<i>E-ticketing</i> dapat berpotensi diintegrasikan di bank atau kartu kredit yang ada
<i>Intermodality</i>	<i>E-ticketing</i> membuat pembayaran untuk perjalanan yang banyak menjadi mudah dan menghasilkan pembayaran yang lebih mudah untuk mendistribusikan melewati mode yang berbeda setelah <i>clearing</i>
<i>Interoperability</i>	<i>E-ticketing</i> membuat pembayaran untuk multi-operator lebih mudah, dan lebih mudah juga untuk mendistribusikan kembali ke operator setelah <i>clearing</i>
<i>Interservices (e-purse)</i>	<i>E-ticketing</i> memungkinkan penggunaan <i>smartcard</i> transportasi umum untuk mendapatkan layanan tambahan yang ditawarkan dalam hubungannya dengan angkutan umum (pembayaran misalnya tempat parkir atau pembelian eceran)
<i>Parking and road pricing</i>	Integrasi dari tol dan tarif parkir dalam kartu yang sama
<i>Customer relationship management (CRM)</i>	<i>E-ticketing</i> adalah alat pemasaran yang kuat karena memungkinkan pengumpulan data rinci tentang perilaku mobilitas pelanggan, yang membantu untuk mengembangkan produk yang ditargetkan
<i>Network monitoring and planning</i>	Data yang dikumpulkan dari <i>ticketing</i> akan meningkatkan pengetahuan tentang <i>boardings</i> dan karena itu memungkinkan untuk kapasitas bus dan jadwal yang disesuaikan dengan penggunaan rute aktual
<i>Secured access and Individual safety</i>	<i>Smartcard</i> juga bisa digunakan sebagai fungsi alarm individu, yang baik menginformasikan pengemudi secara otomatis mentransfer indikasi penumpang ke pusat tanggap darurat

Menerapkan konsep berdasarkan “jarak jangkauan” berikut ada beberapa potensi canggih untuk metode pembayaran

- *Check-in / check-out* (CICO). ini membutuhkan tindakan pengguna secara langsung. Jadi pelanggan harus menunjukkan perangkatnya pada validasi yang terdapat di dalam kendaraan ketika masuk dan atau keluar kendaraan.
- *Walk-in / walk-out* (WIWO), berdasarkan pada antena yang diletakkan pada pintu kendaraan. Mereka masuk dan keluar dengan deteksi dari perangkat yang dibawa oleh user dengan tanpa sebuah tindakan secara khusus
- *Be-in/be-out* (BIBO) system mendeteksi perangkat yang dibawa oleh penumpang ketika kendaraan berpindah dari satu stasiun ke stasiun lain, memungkinkan mendaftar semua penumpang yang sedang berada didalam kendaraan tersebut.

Berdasarkan pemakaian *e-ticketing*, dapat dikumpulkan berbagai informasi, diantaranya:

- Memantau kapasitas dan pembebanan untuk setiap rute
- Dapat memantau bus dan ketepatan waktu kedatangan bus
- Memantau penumpang yang terdapat dalam halte maupun bus
- Perkiraan penumpang per operator dan jenis tiketnya
- Analisis data perjalanan untuk berbagai kelompok penumpang
- Perkiraan on-Demand, waktu, biaya, mode yang terkait dalam perjalanan

Terdapat beberapa jenis tiket elektronik yang sesuai dengan penggunaan nya, diantaranya yaitu tiket tunggal untuk satu atau lebih moda transportasi, tiket asal – tujuan (untuk perjalanan di daerah utama), tiket langganan (dari 1 hari sampai 1 tahun), tiket perjalanan jamak dan tiket yang disesuaikan dengan tarif konsesi.

Tiket langganan mematuhi aturan yang berbeda untuk lokasi yang berbeda. Dalam beberapa kasus mereka valid berdasarkan periode kalender (minggu dari hari Senin sampai Minggu, bulan dari tanggal 1 sampai hari terakhir). Di tempat lain, berlaku mulai hari pertama validasi (7 hari atau 30 hari dari hari pertama validasi). Ada juga kota yang menawarkan lebih banyak fleksibilitas dengan memungkinkan penggunaan kartu 30 hari dalam jangka waktu yang panjang (contoh:

durasi 3 bulan). Tiket langganan umumnya nominatif dan tidak dapat dipindahtangankan. Kartu perjalanan jamak atau buku tiket-jamak dalam beberapa kasus terbatas oleh durasi. Tetapi ada kota yang memperbolehkan tiket perjalanan-jamak tanpa batas waktu, yang membuat mereka gunakan sangat *fleksibel*.

Tiket dengan kemampuan menyimpan “Nilai Uang”, biasanya ditawarkan dalam bentuk tiket *smartcard*. Di Bilbao, Creditrans adalah jenis magnetik tiket yang mampu juga menyimpan nilai.

Tiket Pengunjung yang ditawarkan kepada wisatawan dan biasanya berlaku untuk 7 hari berturut-turut, biasanya lebih mahal daripada tiket langganan dengan durasi yang sama. Yang terakhir umumnya disediakan untuk warga meninggalkan atau bekerja di wilayah tersebut.

Berbasis tarif diskriminasi modus membuat tarif bus yang berbeda dari tarif kereta api untuk satu tiket. Di beberapa kota, pengguna dapat memilih bus-satunya kartu perjalanan atau karcis multimoda. Bila ada lebih dari satu operator bus, tarif satu yang berbeda dapat diterapkan untuk rute bus tergantung operatornya.

## **2.3 Teknologi elektronik dan *smart-card***

Seringkali istilah "kartu *chip*", "kartu sirkuit terpadu" dan "smart card" digunakan secara bergantian, tetapi mereka dapat berarti hal sama atau berbeda. Kartu dibedakan baik oleh jenis chip yang terkandung dan oleh jenis antarmuka (*interface*) yang mereka gunakan untuk berkomunikasi dengan pembaca/ *reader*.

### **2.5.1 Teknologi *Smart-Card***

Ada tiga jenis *chip* yang dapat dikaitkan dengan kartu ini : memori, rangkaian logika dan mikrokontroler. Istilah "memori", "rangkaiannya logika" dan " mikrokontroler" mengacu pada fungsi bahwa *chip* menyediakan :

- ***Memory-Only integrated circuit chip cards***: *Memory-Only* merupakan satunya pengganti kartu "*strip magnetik elektronik*" dan memberikan keamanan sedikit lebih dari kartu *magnetic stripe*. Dua keuntungan yang mereka miliki atas kartu strip magnetik adalah: a ) mereka memiliki kapasitas data yang lebih tinggi (hingga 16 kilobits (kbits) dibandingkan dengan 80 byte per track) dan b) perangkat membaca / menulis yang jauh lebih murah. Kartu memori chip-only tidak mengandung logika atau melakukan

perhitungan, mereka hanya menyimpan data. Kartu *chip* memori Serial dilindungi dan memiliki fitur keamanan yang tidak ditemukan dalam *memori-only chip*, mereka dapat berisi memori tertanam yang tidak dapat ditimpa. Versi awal dari kartu memory-only berkapasitas rendah (maksimal 160 unit value), kartu prabayar sekali pakai dengan sedikit keamanan hanya membaca. Versi baru termasuk kartu prabayar yang menggunakan *read / write memory* dan skema penghitungan biner yang memungkinkan kartu untuk membawa lebih dari 20.000 unit nilai. Banyak dari kartu ini juga memiliki skema otentikasi berbasis logika canggih dibangun ke chip. *Memori-one* yang lain telah dikembangkan untuk *re-loadable* dan menyimpan aplikasi nilai. Kartu yang berisi dompet, dapat dilindungi melalui penggunaan nomor identifikasi pribadi (PIN) dan counter, yang membatasi jumlah kali dompet dapat diisi ulang.

- ***Wired logic integrated circuit chip cards.*** Sebuah *chip wired logic chip*, berisi *state machine* berbasis logika yang menyediakan enkripsi dan akses dikonfirmasi ke memori dan isinya. Kartu *Wired logic* menyediakan sistem file statis yang mendukung beberapa aplikasi, dengan akses terenkripsi (*opsional*) untuk isi memori. File sistem dan set perintah hanya dapat diubah dengan mendesain ulang logika sirkuit terpadu. Kartu *Wired Logic* meliputi variasi *contactless* seperti I-Class atau MIFARE.
- ***Secure microcontroller integrated circuit chip cards.*** Kartu mikrokontroler mengandung mikrokontroler, sistem operasi, dan membaca / menulis memori yang dapat diperbarui berkali-kali. Kartu *chip* mikrokontroler mengandung dan mengeksekusi logika dan perhitungan dan menyimpan data sesuai dengan sistem operasi. Kartu mikrokontroler seperti PC mini yang bisa dibawa dalam dompet. Semua yang dibutuhkan untuk beroperasi adalah kekuatan dan terminal komunikasi. Tersedia jenis kontak dan *contactless* mikrokontroler (*dual interface*). Tidak seperti produk *memori-only*, sirkuit mikrokontroler terpadu ini telah dirancang dengan keamanan yang sangat baik. Dan jenis inilah yang dianggap mulai masuk keluarga *smart-card*.

### 2.5.2 *Memory*

Ukuran memori dinamis pada *smart card* di mana data dapat ditulis atau diubah adalah terbatas, hal itu terjadi akibat pembatasan biaya memori (EEPROM) dan ukuran fisik dari memori *chip* prosesor dalam kartu. Generasi pertama kartu '*read-write*' menawarkan hanya beberapa ratus byte EEPROM. Namun, kartu komersial dengan 4, 8 dan sampai 64K byte sekarang tersedia. Kartu dengan 128K byte juga muncul. 2-4K byte memori cukup untuk menyimpan data balance keuangan dan informasi kontrak, ditambah daftar *auditable* sekitar 100 transaksi terbaru (berisi informasi seperti waktu, lokasi, layanan, biaya dan saldo akhir). Namun, memori benar-benar menentukan berapa banyak fungsi, aplikasi dari kartudan biaya unit kartu.

## 2.4 Protokol Komunikasi E-Ticketing

Secara umum, penggunaan protokol dibagi berdasarkan layer OSI. Berikut ini adalah protokol yang terdapat pada layer OSI :

- *Physical Layer*, terdapat protokol Coaxial, Fiber, Wireless.
- *Data-link Layer*, terdapat protokol Ethernet, 802.11 (Wlan), Wi-fi, Wimax, ATM, Token Ring, Frame Relay.
- *Network Layer*, terdapat protokol Ipv4, Ipv6, IPX, Apple Talk, OSPF, IGMP, ICMP, dan ARPMP.
- *Transport Layer*, terdapat protokol TCP, UDP, SCTP, DCCP.
- *Session Layer*, terdapat protokol NTFS, SQL, RPC, NetBios.
- *Presentation Layer*, terdapat protokol SSL, WPA, WEP.
- *Application layer*, terdapat protokol FTP, POP3, NTP, Tetnet.

Uraian diatas adalah beberapa contoh protokol yang terdapat dalam OSI layer. Dalam sub bab ini, akan dibahas beberapa layer protokol saja, yaitu *data-link layer*, *transport layer* dan *application layer*.

### 2.4.1 *Data-link Layer*

Layer ini berfungsi untuk menentukan bagaimana *bit-bit* data dikelompokkan menjadi format yang disebut *frame*. Selain itu, pada level ini terjadi koreksi kesalahan, *flow control*, pengalamatan perangkat keras dan menentukan bagaimana perangkat-perangkat jaringan beroperasi. Pada layer ini, terdapat beberapa protokol yang akan digunakan untuk membangun protokol *e-ticketing* ini, yaitu 802.3 (*Ethernet*) dan 802.11a/b/g/n.



### 2.4.2 *Transport Layer*

*Layer* ini berfungsi untuk memecah data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Selain itu, pada *layer* ini juga membuat sebuah tanda bahwa paket diterima dengan sukses (*acknowledgement*), dan mentransmisikan ulang terhadap paket-paket yang hilang di tengah jalan. Beberapa protokol yang terdapat pada *transport layer* :

- TCP  
*Transmission Control Protocol* (TCP) adalah salah satu jenis protokol yang memungkinkan kumpulan komputer untuk berkomunikasi dan bertukar data di dalam suatu *network* (jaringan). TCP dipakai untuk aplikasi-aplikasi yang membutuhkan keandalan data. Pada protokol TCP ini, data dikirim ke tujuannya dalam suatu urutan seperti ketika dikirim. Selain itu, TCP juga mempunyai *flow control* yaitu untuk mencegah data terlalu banyak dikirimkan pada satu waktu, yang dapat membuat “macet” jaringan.
- UDP  
*User Datagram Protocol* (UDP) adalah salah satu protokol yang mendukung komunikasi yang tidak andal (*unreliable*) karena pesan-pesan UDP akan dikirim tanpa adanya nomor urut atau pesan *acknowledgement*. Tetapi kelebihan dari UDP ini yaitu UDP adalah salah satu protokol yang ringan, karena protokol ini tidak membutuhkan adanya komunikasi *one-to-one* tetapi *one-to-many*. Jadi pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak bertukar informasi
- SCTP  
*Stream Control Transmission Protocol* (SCTP) adalah suatu protokol yang memiliki beberapa fitur baru seperti *multi-homing*, *multi-streaming* dan *heartbeat*. SCTP adalah suatu protokol yang *connection-oriented*, dimana memerlukan suatu prosedur *call set-up* sebelum terjadi pengiriman data.

### 2.4.3 *Application Layer*

*Layer* aplikasi adalah *layer* yang berfungsi sebagai antarmuka dengan aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan

kesalahan. Protokol yang terdapat pada aplikasi ini adalah FTP, Telnet, dll. Berikut ini akan dijelaskan mengenai beberapa protokol yang terdapat pada *layer* ini

- FTP  
*File Transfer Protocol* (FTP) adalah sebuah protokol internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pengiriman berkas (*file*) komputer antar mesin-mesin dalam sebuah jaringan. FTP menggunakan protokol TCP untuk komunikasi data antara klien dengan *server*, sehingga diantara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum pengiriman data dimulai.
- Telnet  
Telnet adalah terminal interaktif untuk mengakses suatu *remote* pada internet. Fungsi dari telnet ini untuk mengakses *remote host* melalui *terminal* yang interaktif
- HTTP  
*Hypertext Transfer Protocol* (HTTP) adalah protokol yang dipergunakan untuk mentransfer dokumen dalam world wide web

## 2.5 Quality of Service

*Quality of Service* (QoS) dapat dikatakan sebagai suatu terminologi yang digunakan untuk mendefinisikan karakteristik suatu layanan (*service*) jaringan guna mengetahui seberapa baik kualitas dari layanan tersebut. Kualitas layanan diukur dengan standar dari *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) dan standar dari *International Telecommunication Union – Telecommunication* (ITU-T). Dalam penelitian ini parameter QoS yang akan dianalisa adalah *throughput*, *delay*, *jitter* dan *packet loss*.

### 2.5.1 Throughput

*Throughput* diartikan sebagai laju data aktual per satuan waktu (*bits per second*). Biasanya *throughput* selalu dikaitkan dengan *bandwidth*. Karena *throughput* memang bisa disebut sebagai *bandwidth* dalam kondisi yang sebenarnya. *Bandwidth* lebih bersifat tetap sementara *throughput* sifatnya dinamis tergantung *trafik* yang sedang terjadi.

Cara untuk menghitung *throughput* sebagai berikut :

$$throughput = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman}} \quad (1)$$

### 2.5.2 Packet Loss

*Packet loss* menunjukkan jumlah paket yang hilang diantara *node* pengirim dengan *node* tujuan dan diukur dalam *packet loss ratio*. Pengukuran *packet loss* sebagai bahan analisa jaringan pada komunikasi data secara *real time* cukup penting. *Trafik* komunikasi *real time* yang menggunakan transport protokol UDP tidak dapat menjamin sebuah paket data dapat diterima oleh *node* tujuan dengan baik. Berbeda dengan pengiriman paket data menggunakan protokol TCP yang proses pengiriman datanya melalui proses *three-way-handshaking*. Dengan demikian perlu dipastikan kualitas sebuah jaringan untuk komunikasi data *real time*, yang disebut sebagai QoS Untuk menghitung *packet loss* (dalam persen) digunakan rumus berikut:

$$\text{Packet loss rate} = \left( \frac{\text{total packet loss}}{\text{total packet sent}} \right) * 100 \% \quad (2)$$

Berikut ini standar untuk hasil perhitungan *packet loss* versi TIPHON

**Tabel 2.2** Standar *packet loss* versi TIPHON

<i>Packet Loss</i>	Kualitas
0 %	<i>Perfect</i>
3%	<i>Good</i>
15%	<i>Medium</i>
25%	<i>Poor</i>

Sumber : TIPHON

Sedangkan menurut versi ITU-T, terdapat tiga kategori penurunan kualitas jaringan berdasarkan standarisasi nilai *packet loss* sebagai berikut

**Tabel 2.3** Standar *packet loss* versi ITU-T

<i>Packet Loss</i>	Kualitas
3%	Baik
15%	Cukup
25%	Buruk

Sumber : ITU-T G.114

### 2.5.3 Delay

*Delay* didefinisikan sebagai selisih waktu pengiriman sebuah paket saat dikirimkan dengan saat paket tersebut diterima pada *node* tujuan. *Delay* disebut juga dengan istilah *latency* terdiri dari beberapa

faktor penundaan yaitu *propagation delay* atau *transmission delay* yaitu penundaan akibat waktu tempuh paket selama dalam saluran transmisi yang *bandwidth* nya berbeda-deba, *queing delay* yaitu waktu antrian paket sebelum dilewatkan pada saluran transmisi dan lainnya. Waktu tunda dinyatakan dalam satuan detk. Kualitas *delay* dikatakan baik apabila waktu tundanya hanya sekitar 0 – 150 ms (ITU-T). Menurut versi TIPHON standarisasi nilai *delay* sebagai berikut

**Tabel 2.4** Standar *delay* versi TIPHON

Nilai <i>Delay</i>	Kualitas
< 150 ms	<i>Perfect</i>
150 s/d 300 ms	<i>Good</i>
300-450 ms	<i>Medium</i>
> 450 ms	<i>Poor</i>

Sumber : TIPHON

Sedangkan menurut versi ITU-T, standarisasi *delay* sebagai berikut

**Tabel 2.5** Standar *delay* versi ITU-T

Nilai <i>Delay</i>	Kualitas
< 150 ms	Baik
150 s/d 400 ms	Cukup
> 400 ms	Buruk

Sumber : ITU-T G.114

### 2.5.4 *Jitter*

*Jitter* adalah perbedaan selang waktu kedatangan antar paket di terminal tujuan atau dengan kata lain *jitter* merupakan variasi dari *delay*. Besarnya nilai *jitter* mengakibatkan rusaknya data yang diterima, baik itu berupa penerimaan yang terputus-putus atau hilangnya data akibat *overlap* dengan paket data yang lain. Untuk mengatasi *jitter* maka paket yang datang atau melewati sebuah *node* akan diantriakan terlebih dahulu dalam *jitter buffer* selama waktu tertentu hingga nantinya paket dapat diterima pada *node* tujuan dengan urutan yang benar. Namun keberadaan *jitter buffer* akan menambah nilai *end-to end delay*. Berikut ini rumus dari *jitter*

$$Jitter = \frac{\text{Total variasi delay}}{\text{total paket yang diterima}-1} \quad (3)$$

Secara umum terdapat empat kategori penurunan kualitas jaringan berdasarkan versi TIPHON. Berikut ini standarisasi nilai *jitter* versi TIPHON

**Tabel 2.6** Standar *jitter* versi TIPHON

Nilai <i>Jitter</i>	Kualitas
0 ms	<i>Perfect</i>
0 s/d 75 ms	<i>Good</i>
76 s/d 125 ms	<i>Medium</i>
126 s/d 225 ms	<i>Poor</i>

Sumber : TIPHON

## 2.6 Xampp

XAMPP adalah perangkat lunak yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*). Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web *server* Apache, PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasikannya secara otomatis.

Dalam satu paket XAMPP tersedia :

1. Apache Cgi-Bin
2. FTP
3. Mercury Mail (SMTP)
4. PHP
5. MySql
6. Perl
7. PHP Myadmin
8. Webalizer

### 2.6.1 Apache

Tugas utama apache adalah menampilkan halaman web yang benar, sesuai dengan program PHP yang telah dibuat. Apache bersifat *open source*, artinya setiap orang boleh menggunakan, mengambil, dan mengubah kode programnya. Sampai saat ini Apache telah mengalami beberapa perkembangan versi.

### 2.6.2 PHP

PHP (*Personal Home Page*) merupakan bahasa pemrograman yang digunakan untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis, yakni dimana isi informasi website berubah-ubah, dan interaktif dua arah baik dari pemilik maupun pengguna website. PHP dapat dijalankan pada berbagai macam *Operating System*, seperti Windows, Linux, dan Mac OS.

Sistem manajemen *database* yang sering digunakan bersama PHP adalah MySQL. Namun selain itu, PHP juga mendukung sistem manajemen *database* Oracle, Microsoft Access, Interbase, d-Base, PostgreSQL, dan lain-lain. Sama seperti Apache, PHP juga bersifat *open source*.

### 2.6.3 MySQL

SQL merupakan kepanjangan dari *Structured Query Language* yang artinya bahasa terstruktur yang digunakan untuk mengolah *database*. MySQL merupakan sistem manajemen *database* yang bersifat *open source*.

MySQL digunakan untuk membuat dan mengelola *database* beserta isinya, seperti menambahkan, mengubah, dan menghapus data. MySQL juga bersifat relational, artinya data-data yang dikelola akan diletakkan pada beberapa tabel terpisah, sehingga proses manipulasi data akan menjadi lebih cepat.

### 2.6.4 PHPMYAdmin

Salah satu perangkat lunak yang digunakan untuk mengelola *database* dalam MySQL adalah PHPMYAdmin. Dengan PHPMYAdmin kita dapat dengan mudah membuat tabel, mengisi data, dan banyak lagi hal lainnya tanpa harus hafal perintahnya, namun cukup dengan mengisi tabel-tabel yang telah tersedia.

## 2.7 NS2

*Network Simulator* merupakan salah satu perangkat lunak atau *software* yang dapat menampilkan secara simulasi proses komunikasi dan bagaimana proses komunikasi tersebut berlangsung. *Network Simulator* melayani simulasi untuk komunikasi dengan kabel dan komunikasi *wireless*.



*Network Simulator* dibangun dengan menggunakan 2 bahasa pemrograman, yaitu C++ dan Tcl/OTcl. C++ digunakan untuk *library* yang berisi *event scheduler*, protokol dan *network component* yang diimplementasikan pada simulasi oleh user. Tcl/OTcl digunakan pada *script* simulasi yang ditulis oleh NS user dan pada *library* digunakan sebagai *simulator* objek. Otcl nantinya juga berperan sebagai interpreter.

Bahasa C++ digunakan pada *library* karena C++ mampu mendukung *runtime* simulasi yang cepat, meskipun simulasi melibatkan simulasi jumlah paket dan sumber data dalam jumlah besar. Sedangkan bahasa Tcl memberikan respon *runtime* yang lebih lambat daripada C++, namun jika terdapat kesalahan, respon Tcl terhadap kesalahan *syntax* dan perubahan *script* berlangsung dengan cepat dan interaktif.

Berikut ini adalah script yang digunakan pada NS2 untuk menambahkan link antar node.

```
$ns <tipelink> $src $dst <bw> <delay> <tipecqueue>
```

Contoh script dalam simulasi :

```
$ns duplex-link $node0 $node1 100Mb 10ms DropTail
```

Keterangan:

<tipelink>, diantaranya :

Simplex-link : Komunikasi searah

Duplex-link : Komunikasi dua arah

<bw>, merupakan link *bandwidth* pada sebuah kanal jaringan.

<delay>, merupakan link delay pada sebuah kanal jaringan.

*Delay* disini merupakan ukuran jarak yang sebenarnya antara *node* satu dengan *node* yang lain. Jarak yang sebenarnya pada lapangan akan dikonversi menjadi *delay* sehingga dapat di simulasikan pada *software* NS2, dengan menggunakan rumus

$$D = T \cdot V \quad (4)$$

Keterangan

D = Jarak (meter)

T = Waktu perjalanan gelombang (s)

V = kecepatan cahaya (m/s)

Hasil dari perhitungan rumus diatas, terdapat pada lampiran

## **2.8 Sistem *E-Ticketing* di Berbagai Kota Besar di Dunia**

Kajian mengenai sistem *e-ticketing* dilakukan di berbagai negara. Kajian ini merupakan tahap melihat dan mengamati tentang proses sistem *e-ticketing* di kota-kota besar seperti di kota London, kota Tokyo dan Singapura. Berikut ini penjelasan mengenai sistem *e-ticketing* di berbagai kota besar di dunia

### **2.8.1 London**

Kota London menggunakan tiket *oyster card* sebagai tiket elektronik nya yang digunakan sebagai alat pembayaran yang sah untuk melakukan perjalanan menggunakan moda transportasi yang ada. Moda transportasi tersebut adalah *london underground*, *london buses*, *the docklands light railways* (DLR), *london overground*, *trams*, beberapa *river boat service*, dan kereta nasional. Standar ukuran dari tiket ini yaitu sebesar kartu kredit dengan warna biru yang didalam nya dapat memuat satu tiket, tiket periodik, dan tiket sesuai permintaan. Tiket tersebut harus ada ditangan penumpang sebelum berpergian. Penumpang menempelkan tiket itu ke pembaca ketika memasuki dan keluar dari sistem transportasi. Kartu tersebut dapat dilakukan pengisian ulang saldonya dengan cara pembelian saldo secara online maupun secara tunai.

Untuk penggunaan tiket ini, penumpang diharuskan melakukan proses *tap-in* dan *tap-out*. Sewaktu penumpang ingin memasuki kendaraan, pada beberapa halte terdapat penghambat/penghalang yang mewajibkan penumpang melakukan proses *tap-in* di tempat tersebut. tetapi pada beberapa halte yang kecil, penghalang tersebut tidak ada, sehingga penumpang diharuskan melakukan proses *tap-in* di dalam kendaraan.

Ketika saldo pada tiket mulai sedikit, saldo dapat diisi pada halte/stasiun yang menyediakan alat pengisian saldo. Selain itu penumpang juga mendapat pilihan, yaitu penumpang dapat memilih pengisian saldo secara langsung (*auto top-up*). Sistem ini berjalan ketika saldo penumpang dibawah harga tiket, dan sistem secara otomatis akan menambahkan saldo ke tiket tersebut.

### **2.8.2 Tokyo, Jepang**

Kota Tokyo menggunakan tiket dengan nama Suica. Kegunaan utama dari suica adalah sebagai kartu tarif untuk jasa transportasi, selain itu kartu ini juga dapat digunakan sebagai uang elektronik untuk pembelian umum. Kebanyakan mesin penjual, kios dan loker dapat

dioperasikan menggunakan kartu tersebut. Selain pembayaran, kartu tersebut juga dapat digunakan sebagai kunci elektronik untuk membuka loker khusus. Kartu ini juga dapat digunakan untuk melakukan pembayaran online, yang mengharuskan pengguna mempunyai perangkat keras Sony FeliCa *Reader* dan PC yang menjalankan sistem operasi Windows.

Cara penggunaan dari tiket ini yaitu, penumpang terlebih dahulu membeli kartu ke mesin deposit kartu. Penumpang diwajibkan memasukkan uang sebesar 2000 yen, 500 yen untuk biaya pembuatan kartu, dan sisanya sebagai deposit saldo kita. Sebelum memasuki kendaraan, penumpang diwajibkan untuk melakukan proses *tap* ke salah satu gerbang yang terdapat tanda suica/ic. Setelah penumpang turun dari kendaraan, penumpang diwajibkan melakukan *tap* lagi sebelum keluar dari terminal/halte

Jika saldo habis, penumpang dapat mengisi saldo tiket pada mesin pengisian saldo dan bisa juga mengisi di minimarket. Selain itu, uang yang terdapat pada tiket tersebut dapat diuangkan kembali jika sudah selesai dipergunakan. Namun, jika saldo ternyata habis tetapi kita tidak mengetahui bahwa saldo tiket telah habis, maka pada saat akan melakukan proses *tap* pada gerbang, gerbang tidak akan terbuka dan diharuskan mengisi saldo terlebih dahulu.

### **2.8.3 Singapura**

Kota Singapura menggunakan tiket bernama Ez-Link yang berbasis kepada Sony FeliCa *smart card technology* dan digunakan untuk pembayaran biaya transportasi publik. Kartu ini digunakan untuk pembayaran *mass rapid transport* (MRT), *Light Rail Transit* (LRT) dan *public bus service*.

Cara penggunaan dari ez-link ini, yaitu setelah penumpang mendapatkan kartu ini, kemudian penumpang dapat mengecek saldo di mesin tiket yang terdapat di stasiun. Jika dirasa saldo tersebut kurang, maka penumpang dapat menambah saldo langsung pada mesin tersebut. Bila saldo kartu kurang dari \$3, kartu tidak dapat digunakan. Untuk penggunaan dalam pembayaran perjalanan cukup sederhana, yaitu sebelum naik kendaraan, penumpang diwajibkan melakukan proses *tap-in* pada gerbang masuk, setelah itu jika penumpang telah melakukan perjalanan, maka penumpang diwajibkan melakukan *tap-out* pada mesin yang terdapat di gerbang (*gate*) agar biaya perjalanan dapat dihitung dengan sempurna

## 2.9 Security Tiket Elektronik

Keamanan dari sebuah sistem kartu pintar (*smartcard*) terdiri dari keamanan depan (*front-end*), verifikasi depan dan belakang (*front-end and back-end*) dan audit sistem terakhir (*back-end*). Sistem keamanan yang didesain secara benar, dapat meminimalisir adanya penipuan, penipuan hanya mungkin dilakukan antara pemegang kartu dari pedagang/pengelola.

Keamanan *smartcard* lebih banyak menggunakan sistem keamanan MIFARE. Sistem keamanan ini ada beberapa jenis, diantaranya yaitu MIFARE *Classic*, MIFARE *Ultralight*, MIFARE DESFire, MIFARE *Plus*, MIFARE SAM AV2.

Nama MIFARE diambil dari istilah *Mikron FARE Collection System*, meliputi tujuh jenis kartu yang berbeda, yaitu :

- MIFARE *Klasik*  
Menggunakan sebagian protokol khusus dari ISO/IEC 14443-3 *type a*, dengan protokol NXP yang digunakan untuk autentikasi dan pengkodean
- MIFARE *Ultralight*  
Termasuk ke dalam IC yang mempunyai biaya rendah yang menggunakan protokol khusus sesuai dengan ISO/IEC 14443-3 *type a*
- MIFARE Ultralight C  
IC murah yang menggunakan aplikasi dari pengkodean *Triple DES*
- MIFARE DESFire  
Smartcard yang menggunakan standar ISO/IEC 14443-3 *type A* dengan sistem operasi *mask-ROM* dari NXP
- MIFARE DESFire EV1  
Menggunakan standar enkripsi AES
- MIFARE DESFire EV2  
Termasuk *MismartApp*, *Transaction MAC*, *Unlimited Application*
- MIFARE *Plus*  
Penggantian dari MIFARE klasik dengan level keamanan bersertifikat (AES 128 *Based*)
- MIFARE SAM AV2  
Modul akses yang aman yang menyediakan penyimpanan dari kriptografi dan fungsi dari kriptografi

Berikut ini penjelasan perbagian dari berbagai jenis MIFARE

### **MIFARE Klasik**

Kartu MIFARE klasik ini dasarnya hanya perangkat penyimpanan memori, dimana memori dibagi menjadi *segmen* dan blok dengan mekanisme keamanan sederhana untuk kontrol akses. Dengan berbasis ASIC yang dapat digunakan untuk pembatasan komputasi dan mempunyai biaya rendah sehingga kartu ini banyak digunakan untuk dompet elektronik, kontrol akses, kartu ID perusahaan, transportasi atau stadion *ticketing*

MIFARE klasik 1K menawarkan 1024 *byte* penyimpanan data, dibagi menjadi 16 sektor, masing-masing sektor dilindungi oleh dua kunci yang berbeda, yang disebut A dan B. Setiap tombol dapat diprogram untuk memungkinkan operasi seperti membaca, menulis, meningkatkan blok nilai, dll.

MIFARE klasik 4K menawarkan 4096 *byte* yang dibagi menjadi empat puluh sektor, yang 32 merupakan sektor yang sama seperti sektor 1K dengan delapan sektor lagi yang mempunyai ukuran empat kali lipat.

### **MIFARE Ultraligh EV1**

Diperkenalkan pada bulan November 2012, generasi terbaru dari *smartcard* yang menawarkan pengembangan solusi dan keamanan tambahan. Jenis ini muncul dengan beberapa perangkat tambahan seperti

- 384 dan 1024 varian produk BITS memori pengguna
- OTP, *Lock bits*, *counter* dikonfigurasi untuk meningkatkan keamanan
- Dilindungi akses data melalui sandi 32-bit
- NXP *semiconductors* dengan fungsi *signature* orisinalitas, ini adalah pengecekan secara terpadu dan merupakan perlindungan terhadap kloning yang paling efektif.

## BAB III

### PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan tahap perancangan dan implementasi sistem *e-ticketing*. Pengerjaan yang pertama yaitu perancangan sistem *e-ticketing* secara umum, dari proses pembacaan tiket ke OBU, OBU ke *Access Point* yang berada di halte, sampai dengan data dapat diterima pada *server*. Setelah itu, perancangan umum tersebut akan diterapkan pada Angkutan Masal Cepat (AMC) Kota Surabaya. Pada pengerjaan tugas akhir ini, akan dibuat sistem pentarifan yang cocok untuk di implementasikan pada kota ini. Setelah pembuatan tarif selesai, maka dilakukan implementasi sistem dan membuat skenario untuk pengujian sistem tersebut.

#### 3.1 Perancangan Sistem *E-Ticketing*

Sistem *e-ticketing* secara umum merupakan alur kerja pembacaan kartu (*smartcard*) yang didalam nya terdapat berbagai macam informasi sampai dengan data yang terdapat pada kartu tersebut sampai di *server*. Gambar 3.1 merupakan gambaran secara umum mengenai sistem *e-ticketing*



**Gambar 3.1** Skema *e-ticketing* secara umum

Gambar diatas menjelaskan berlangsungnya komunikasi dari kartu sampai dengan *server*. Berikut ini akan dijelaskan mengenai proses komunikasi tersebut

##### 3.1.1 Proses Komunikasi Tiket – OBU

Komunikasi paling awal dari *e-ticketing* adalah pembacaan data yang terdapat di dalam kartu (tiket). Data tersebut berupa ID yang sudah tersimpan di database. ID yang terdapat di database tersebut memuat



beberapa informasi diantaranya saldo, identitas pemilik, dan lain-lain. Kartu yang digunakan pada tahap ini adalah kartu yang berjenis *near Field Communication* (NFC). Kartu tersebut berupa kartu dengan sandart ISO 14443 yang mampu dikenai tindakan *read* / *write* dan mampu menyimpan berbagai jenis data dan beroperasi pada rentang yang berbeda.

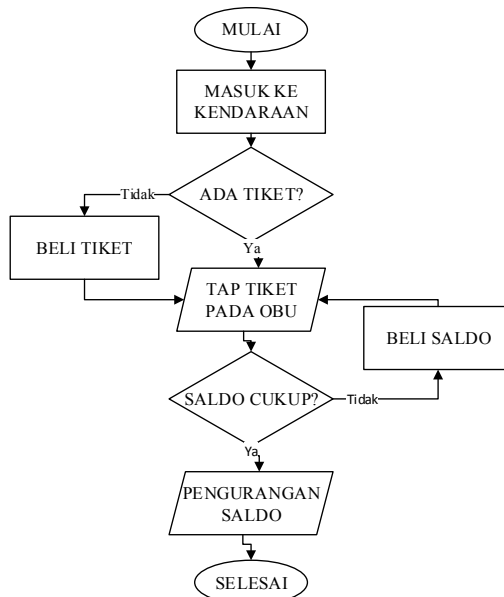


**Gambar 3.2** Komunikasi dari tiket ke OBU

Pada gambar 3.2 diatas, menunjukkan bahwa adanya komunikasi antara tiket dengan pembaca atau OBU. Pada tahap awal pembuatan protokol *e-ticketing* ini, dibuat suatu skenario yaitu bagaimana user menggunakan tiket untuk melakukan pembayaran suatu perjalanan. User masuk ke dalam kendaraan, kemudian melakukan *tap-in* ke OBU. Jika user tidak mempunyai tiket, maka user harus membeli tiket terlebih dahulu. Tetapi jika user sudah mempunyai tiket, user dapat langsung melakukan proses *tap-in*. Sewaktu melakukan *tap-in*, ternyata saldo user tidak mencukupi, maka user diwajibkan untuk mengisi saldo terlebih dahulu. Di dalam kendaraan, terdapat mesin deposit, sehingga user dapat langsung mengisi saldo pada mesin tersebut. Setelah dipastikan user mempunyai saldo, maka pada saat user melakukan *tap-in* akan terjadi pengurangan saldo.

Pada saat proses *tap-in* dilakukan, OBU membaca data yang terdapat di dalam kartu. Salah satu data tersebut adalah nomor kartu (id kartu) dan saldo kartu. Sehingga pada saat proses *tap-in* saldo yang terdapat pada tiket akan langsung dipotong oleh OBU, selanjutnya OBU akan menulis (*write*) jumlah saldo hasil dari pemotongan tersebut ke tiket milik user.

Proses tersebut digambarkan pada gambar 3.3 dibawah ini.



**Gambar 3.3** Flowchart proses *tap-in*

### 3.1.2 Proses Komunikasi OBU – Halte

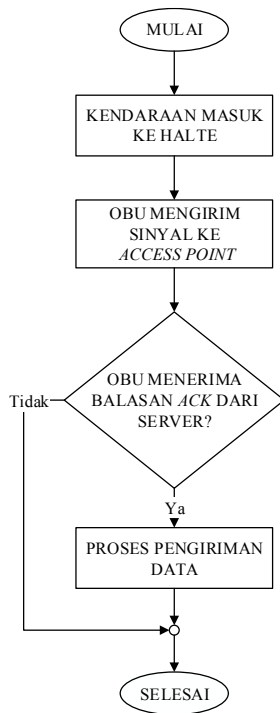
Komunikasi berikutnya adalah komunikasi dari OBU ke halte. Pada komunikasi ini, data yang tersimpan sementara di OBU akan dikirim ke *server*. Sebelum di kirim ke *server*, data harus melewati halte sebagai penghubung antara *server* dengan OBU. Pengiriman data ini dilakukan dengan perangkat *wireless access point*.



**Gambar 3.4** Komunikasi dari OBU ke *Acess Point*

Gambar 3.4 diatas adalah gambar ilustrasi komunikasi antara OBU dengan halte. Pada gambar tersebut, data yang terdapat pada OBU akan di kirim ke halte melalui *access point*. Sebelum terjadinya proses

pengiriman data, kendaraan harus masuk ke halte terlebih dahulu. Di halte, terdapat *access point* yang digunakan sebagai media komunikasi antara OBU dengan *server*. OBU yang terletak di kendaraan akan mengirimkan sinyal ke sembarang arah. Pada saat pengiriman sinyal tersebut, *access point* mendeteksi adanya sinyal dari OBU. Selanjutnya, *access point* memberikan *acknowledgement* pada OBU bahwasanya sinyal dari OBU dapat diterima dengan baik oleh *access point*. Setelah OBU menerima sinyanya ACK tersebut, OBU mulai mengirimkan data ke *access point*. Tetapi, jika sewaktu OBU memancarkan sinyal ke sembarang arah, tetapi tidak mendapat balasan dari *access point*, maka data sementara akan disimpan di dalam OBU terlebih dahulu. Berikut ini gambar 3.5 yang menggambarkan terjadinya komunikasi dari OBU ke Halte.



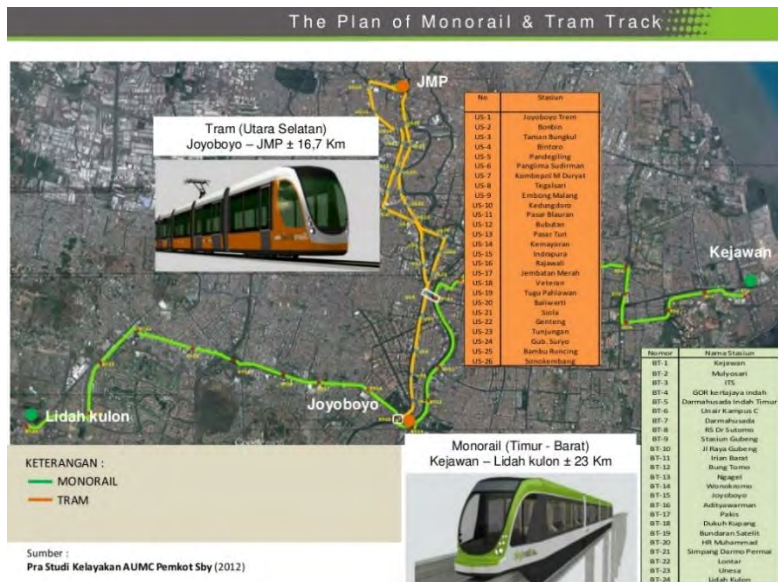
**Gambar 3.5** Flowchart komunikasi dari OBU ke Halte

### 3.1.3 Proses Komunikasi Halte – Server

Data yang sudah sampai di *access point*, selanjutnya dikirim ke *server*. Halte hanya berfungsi sebagai perantara antara OBU dengan *server*. Oleh karena itu, data yang dikirim dari OBU mempunyai *header*/alamat tujuan ke *server*. Data yang sudah masuk di *access point* selanjutnya dikirim ke alamat tujuan yaitu *server e-ticketing*. Setelah data terkirim ke *server* selanjutnya data tersebut akan disimpan di *server* dan akan diolah dalam sistem *e-ticketing*.

## 3.2 AMC Kota Surabaya

Angkutan Masal Cepat (AMC) juga diterapkan di Kota Surabaya. Pada AMC Surabaya ini, terdapat beberapa moda transportasi yang digunakan seperti angkot, minibus, bus, tram dan monorail. Dalam pengerjaan tugas akhir ini, penulis coba memfokuskan pada salah dua moda transportasi saja, yaitu monorail dan tram. Berikut ini gambar 3.6 yang merupakan gambaran dari jalur monorail dan tram



**Gambar 3.6** Jalur Monorail dan Tram

Angkutan masal diatas biasa disebut sebagai Suratrām dan Boyorail. Kedua jenis moda transportasi ini, menghubungkan Surabaya bagian timur-barat dan utara-selatan. Surotram menghubungkan Surabaya bagian utara sampai Surabaya bagian selatan dengan total jarak  $\pm 16,7$  Km dan melewati 26 titik halte. Daerah yang dilewati oleh surotram yaitu Sonokembang, Bambu Runcing, Gub. Suryo , Tunjungan, Genteng, Siola, Baliwerti, Tugu Pahlawan, Veteran, Jembatan merah, Rajawali, Indrapura, Kemayoran, Psar Turi, Bubutan, Pasar Blauran, Kedungdoro, Embong Malang, Tegalsari, Kombepol M Duryat, Panglima Sudirman, Pandegiling, Bintoro, Taman Bungkul, Bonbin dan Joyoboyo. Boyorail menghubungkan Surabaya bagian timur sampai Surabaya bagian barat dengan total jarak  $\pm 23$  Km dan melewati 24 titik halte. Daerah yang dilewati oleh boyorail yaitu Kejawan, Mulyosari, ITS, Gor Kertajaya, Dharmahusada Indah Timur, RS DR Sutomo, Stasiun Gubeng, Jl Raya Gubeng, Irian Barat, Bung Tomo, Ngagel, Wonokromo, Joyoboyo, Adityawrman, Pakis, Dukuh Kupang, Bundaran satelit, HR Muhammad, Simpan Darmo Permai, Lontar, Unesa, Lidah Kulon.

### 3.3 Sistem Pentarifan *E-Ticketing*

Untuk sistem *e-ticketing* di Surabaya, dilakukan pengelompokan besaran tarif yang harus dibayar oleh penumpang sesuai dengan klasifikasi sebagai berikut (diasumsikan semua perjalanan dengan tarif rata sebesar 5000 rupiah). Berikut ini tabel 3.1 yang merupakan klasifikasi pentarifan e-ticketing

**Tabel 3.1** Klasifikasi pentarifan e-ticketing

Kelompok	Kepala Tiket	Diskon		Harus Dibayar
		%	Rupiah	
Anak – Anak atau pelajar	1xxxx	50	2500	2500
Dewasa atau pekerja	2xxxx	0	0	5000
Orang Tua atau Pensiunan	3xxxx	25	1250	3750
Difabel	0xxxx	50	2500	2500

Dalam tugas akhir ini, dibuat contoh seorang pelajar akan menggunakan moda transportasi monorail. Pelajar tersebut naik di halte bundaran ITS dengan tujuan ke halte Joyoboyo. Agar pelajar tersebut dapat sampai ke tujuan, maka dia harus :

1. Mempunyai tiket terlebih dahulu. Jika dia tidak mempunyai tiket, maka dia dapat membeli karcis sewaktu berada di dalam monorail. Namun jika dia mempunyai tiket, dia dapat langsung naik ke dalam kendaraan
2. Setelah pelajar tersebut masuk ke dalam kendaraan, langkah selanjutnya yaitu melakukan proses *tap-in*. Pada proses ini, setelah pelajar masuk di dalam monorail, pelajar mendekati OBU (*reader*) dan menempelkan kartu (tiket/*smartcard*). Jika saldo mencukupi, maka perangkat tersebut akan mengeluarkan karcis sebagai bukti bayar bahwa dia telah melakukan proses *tap-in*. Jika saldo tidak mencukupi, maka dia dapat membayar biaya perjalanan secara tunai di mesin yang telah disediakan
3. Setelah proses *tap-in* berhasil, langkah selanjutnya yaitu menunggu sampai dengan kendaraan monorail sampai di halte Joyoboyo.
4. Setelah kendaraan monorail sampai di halte Joyoboyo, langkah selanjutnya yaitu melakukan proses *tap-out*. Proses ini dilakukan dengan cara menempelkan kartu pada mesin pembaca yang berada di sebelah pintu keluar. Proses ini wajib dilakukan agar pintu keluar kendaraan dapat terbuka, selain itu proses ini ditunjukkan agar memudahkan proses perhitungan penumpang yang masih berada di dalam kendaraan.

### **3.4 Implementasi Sistem *E-ticketing***

Pada pengerjaan tugas akhir ini, sistem yang telah dibuat akan di implementasikan. Implementasi sistem yang pertama yaitu implementasi sistem dengan software Xampp. Sistem dibuat dengan versi *website* yang didalamnya terdapat proses *tap-in*, *tap-out*, dll. Implementasi yang kedua yaitu implementasi jaringan. Pada implementasi ini, jaringan akan dibuat dengan software NS2 mengikuti rute/halte yang terdapat pada rancangan AMC Kota Surabaya

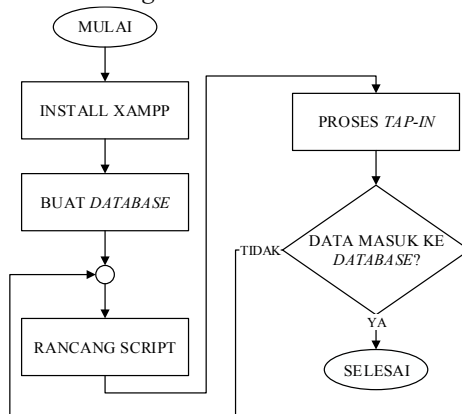
#### **3.4.1 Implementasi Sistem**

Pada pengerjaan tugas akhir ini, implementasi sistem dilakukan menggunakan server lokal dengan memanfaatkan database yang sudah



ada pada software Xampp. Software ini memiliki fungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP *server*, Mysql *database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman php dan perl.

Sistem *e-ticketing* di implementasikan di *localhost*, yang pertama kali dilakukan yaitu *install software* Xampp, setelah proses instalasi selesai, selanjutnya membuat *database* pada PHPmyadmin. *Database* tersebut berisikan beberapa tabel, diantaranya tabel kartu, tabel kendaraan, tabel rekapitulasi dan tabel halte. Setelah *database* berhasil dibuat, selanjutnya merancang *script* yang digunakan untuk simulasi sistem *e-ticketing*. *Script* yang dibuat yaitu *script tap-in*, *script tap-out*, *script* monitoring dan *script* koneksi ke *database*. Setelah semua *script* selesai di buat, selanjutnya dicek apakah *script* tersebut telah berfungsi sebagaimana mestinya atau tidak. Pengecekan ini dengan cara melakukan proses *tap-in* pada *browser*, jika data berhasil masuk ke *database* maka *script* yang dirancang telah berfungsi sebagaimana mestinya, tetapi jika data tidak masuk ke dalam *database*, maka harus dilakukan pengecekan ulang pada *script* yang telah dibuat. Berikut gambar 3.7 yang merupakan pengerjaan sistem *e-ticketing*



**Gambar 3.7** Flowchart pembuatan sistem e-ticketing

Dari penjabaran diatas, langkah pertama pengerjaan yaitu menginstall Xampp. Jika Xampp sudah terinstall maka selanjutnya membuat *database*. *Database* dibuat sesuai dengan format perancangan

sistem *e-ticketing*. Database diberi nama 'simulasi'. Tabel yang terdapat pada database yaitu tabel kendaraan, tabel tiket, tabel rekap dan tabel halte. Berikut ini gambar 3.8 yang merupakan isi dari database simulasi



**Gambar 3.8** Daftar database

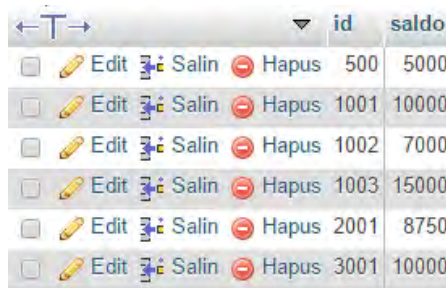
Fungsi dari tabel kendaraan yaitu untuk mencatat laporan langsung dari kendaraan. Laporan langsung tersebut berupa laporan dimana kendaraan sekarang berada, selain itu terdapat jumlah penumpang yang sedang berada di dalam kendaraan tersebut. Berikut ini gambar 3.9 yang merupakan tampilan isi dari tabel kendaraan

			id	kend	latitude	longitude	total	waktu
			1	monorail	-7.277030	112.803186	1	2016-01-05 07:34:52
			2	tram	-7.236493	112.737597	1	2016-01-05 06:43:35

**Gambar 3.9** Tampilan tabel kendaraan

Gambar diatas adalah tampilan isi tabel kendaraan. Tabel tersebut berisikan 2 baris saja, sesuai dengan kendaraan dalam perancangan sistem *e-ticketing* ini, yaitu kendaraan dengan id 1 yang berarti monorail dan kendaraan dengan id 2 yang berarti tram. Setelah kolom kend terdapat kolom *latitude* dan kolom *longitude* yang berfungsi untuk merekam posisi kendaraan pada sewaktu terjadi perubahan data. Setelah kolom tersebut, terdapat kolom total. Kolom total berfungsi sebagai pencatat jumlah penumpang yang sedang berada di dalam kendaraan tersebut, dan yang terakhir adalah kolom waktu. Kolom waktu berfungsi sebagai pencatat waktu ketika terjadi perubahan data pada tabel kendaraan.

Selain tabel kendaraan, terdapat tabel tiket yang berfungsi untuk mencatat nomor tiket/id tiket dan saldo. Berikut ini gambar 3.10 yang merupakan tampilan isi dari tabel tiket



					id	saldo
<input type="checkbox"/>		Edit		Salin		Hapus
					500	5000
<input type="checkbox"/>		Edit		Salin		Hapus
					1001	10000
<input type="checkbox"/>		Edit		Salin		Hapus
					1002	7000
<input type="checkbox"/>		Edit		Salin		Hapus
					1003	15000
<input type="checkbox"/>		Edit		Salin		Hapus
					2001	8750
<input type="checkbox"/>		Edit		Salin		Hapus
					3001	10000

**Gambar 3.10** Tampilan tabel tiket

Setelah tabel tiket, terdapat tabel rekap. Tabel rekap berfungsi sebagai penyimpanan semua kegiatan selama sistem *e-ticketing* berlangsung. Berikut ini gambar 3.11 yang merupakan tampilan isi dari tabel rekap


<div><div><div></div><div></div></div><div></div></div>											id	kend	kartu	latitude	longitude	waktu	userin	userout
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	3	tram	500	-7.246518	112.737003	2015-12-28 20:36:21	0	1					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	4	tram	500	-7.237673	112.731037	2015-12-28 20:38:26	0	1					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	5	tram	500	-7.236493	112.737597	2015-12-28 20:38:51	0	1					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	6	tram	500	-7.298569	112.737869	2015-12-28 20:41:24	1	0					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	7	tram	500	-7.236493	112.737597	2015-12-28 20:41:52	0	1					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	8	tram	500	-7.246507	112.738660	2015-12-28 20:42:54	0	1					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	9	tram	2001	-7.237673	112.731037	2015-12-28 21:56:22	1	0					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	10	tram	2001	-7.236493	112.737597	2015-12-28 21:57:10	0	1					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	11	monorail	1002	-7.298378	112.737783	2015-12-28 22:03:15	1	0					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	12	monorail	1002	-7.300095	112.662354	2015-12-28 22:03:48	0	1					
<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	<div><div><div></div><div></div></div><div></div></div>	Salin	<div><div><div></div><div></div></div><div></div></div>	Hapus	13	tram	500	-7.236493	112.737597	2016-01-05 06:43:36	1	0					

**Gambar 3.11** Tampilan tabel rekap

Pada tabel rekap, terdapat beberapa kolom. Kolom yang pertama adalah kolom ID untuk mewakili id setiap baris data. Kolom yang kedua adalah kolom kend yang berfungsi sebagai pencatat tipe atau jenis kendaraan pada saat sistem berlangsung. Kolom tersebut baru berisi kendaraan dengan tipe monorail dan tram, sesuai dengan tabel kendaraan. Kolom selanjutnya yaitu RFID, kolom ini mengacu pada tabel tiket.

Kolom selanjutnya yaitu kolom *latitude* dan *longitude* yang berfungsi sebagai pencatat koordinat atau lokasi disaat komunikasi terjadi. Setelah itu ada kolom waktu, kolom *userin* dan kolom *userout*. Kolom waktu berfungsi untuk mencatat waktu pada saat proses *tap-in* maupun *tap-out* terjadi. Kolom *userin* dan *userout* berfungsi sebagai penanda disaat user melakukan proses *tap-in* maupun proses *tap-out*.

Setelah tabel rekap, terdapat tabel halte. Tabel halte berfungsi sebagai pencatat koordinat setiap halte. Halte dibagi menjadi 2 jalur, jalur pertama yaitu jalur monorail dan jalur kedua merupakan jalur tram. Berikut gambar 3.12 yang merupakan tampilan dari tabel halte. Tabel halte yang lengkap terdapat pada lampiran



			id	halte	lat	lon	jalur
<input type="checkbox"/>	Edit	Salin	Hapus	16	Adityawarman	-7.293546	112.730595 monorail
<input type="checkbox"/>	Edit	Salin	Hapus	44	Baliwerti	-7.252513	112.736791 tram
<input type="checkbox"/>	Edit	Salin	Hapus	49	Bambu Runding	-7.267846	112.744561 tram
<input type="checkbox"/>	Edit	Salin	Hapus	28	Bintoro	-7.286352	112.739693 tram
<input type="checkbox"/>	Edit	Salin	Hapus	26	Bonbin	-7.29559	112.739264 tram
<input type="checkbox"/>	Edit	Salin	Hapus	36	Bubutan	-7.2507	112.735766 tram
<input type="checkbox"/>	Edit	Salin	Hapus	19	Bundaran Satelit	-7.287523	112.704459 monorail
<input type="checkbox"/>	Edit	Salin	Hapus	12	Bung Tomo	-7.288942	112.744762 monorail
<input type="checkbox"/>	Edit	Salin	Hapus	7	Darmahusada	-7.266281	112.766322 monorail
<input type="checkbox"/>	Edit	Salin	Hapus	5	Dharmahusada Indah Timur	-7.275157	112.781664 monorail

**Gambar 3.12** Tampilan tabel halte

Setelah *database* berhasil dibuat, langkah selanjutnya yaitu membuat *script* untuk simulasi *e-ticketing*. *Script* yang dibuat diantaranya adalah *script* untuk proses *tap-in*, proses *tap-out*, proses tambah saldo, *script* monitoring dan *script* untuk koneksi ke *database*. Untuk memudahkan dalam membuat simulasi ini, tabel yang berada di database di tampilkan pada web browser. Tampilan tersebut yaitu tampilan tabel kartu, tampilan tabel kendaraan serta tampilan tabel rekap. Pada simulasi ini, dibuat alur menyerupai dengan proses tap-in pada kondisi yang sebenarnya.

Alur tersebut yaitu masuk ke halaman Update Live, setelah itu akan tampil daftar kendaraan yang sedang beroperasi, dalam simulasi ini, penulis hanya mendaftarkan satu kendaraan monorail, dan satu kendaraan tram. Setelah itu, menuju ke bagian *tap-in*. Pada bagian tersebut, terdapat

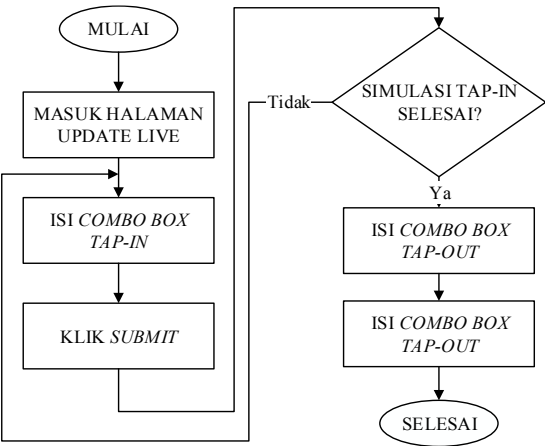
*combo box* yang harus diisi, *combo box* tersebut yaitu kartu, kendaraan dan halte. Kartu pada proses ini adalah kartu yang sudah terdaftar pada *database*. Pada *combo box* kendaraan terdapat 2 pilihan, yaitu monorail dan tram, selanjutnya pada *combo box* terakhir, tampil daftar halte. Daftar halte yang tampil tersebut sesuai dengan kendaraan yang dipilih, jika memilih monorail, maka yang akan tampil adalah daftar halte yang dilewati oleh kendaraan tersebut, begitu juga dengan tram. Daftar halte disini digunakan untuk mengecek posisi transaksi terakhir ada di halte mana. Jika semua *combo box* sudah terisi, selanjutnya klik *submit*.

Tombol *submit* tersebut mengarahkan sistem menuju *script tap-in* yang berisi pemotongan saldo berdasarkan *range* tiket yang telah dirancang, *update* saldo terbaru pada tabel tiket, membuat baris baru pada

tabel rekap, *update* total user yang berada di dalam kendaraan dan *update* lokasi kendaraan pada saat ini.

*Script* yang lain yaitu *script tap-out*. Isi dari *script* ini yaitu mengurangi jumlah user yang berada di dalam kendaraan dan *update* lokasi kendaraan pada saat proses *tap-out* berlangsung.

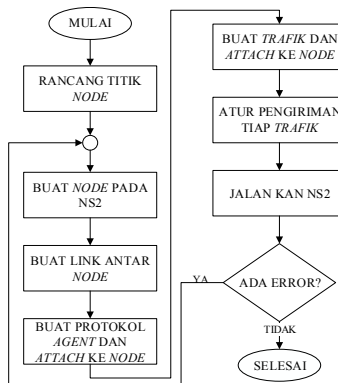
Berikut ini gambar 3.13 yang merupakan flowchart dari simulasi sistem *e-ticketing*



**Gambar 3.13** Flowchart simulasi sistem *e-ticketing*

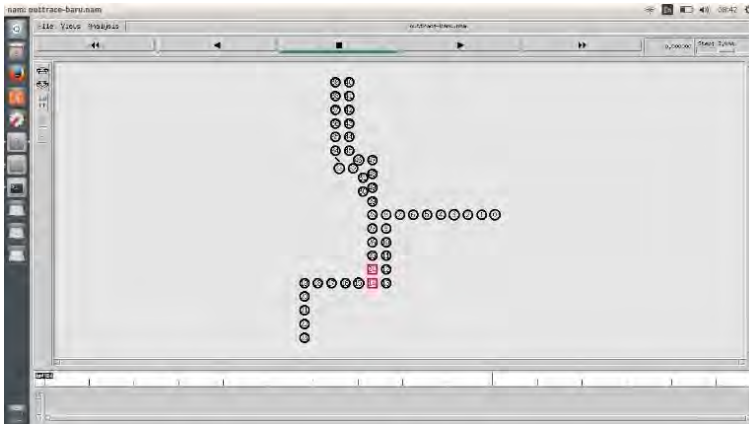
### 3.4.2 Implementasi Jaringan

Pada perancangan tugas akhir ini, akan disimulasikan jaringan hasil dari rancangan sebelum nya. Simulasi ini dilakukan pada *software* NS2 yang beroperasi pada sistem operasi Ubuntu versi 14.04 LTS. Proses pembuatan simulasi ini, dimulai dari perancangan titik *node*. Titik *node* dirancang mengikuti rancangan halte tram dan monorail di Kota Surabaya. Setelah di dapat titik lokasi per *node* dan di dapat jarak antar *node*, selanjutnya dibuat *node* pada *software* NS2. Total *node* pada perancangan ini yaitu 50 *node* dengan 2 *server* dan 2 jalur. Setelah itu dibuat *link* antar *node*. *Link* yang dibuat berjenis *duplex-link*. Setelah itu dibuat protokol *agent* kemudian *attach* ke *node* dan buat *trafik* kemudian *attach* ke *agent*. *Agent* yang dibuat yaitu TCP dengan *trafik* FTP. *Trafik* FTP di set sebesar 100 kb pada link pengirim tiap *node* dengan tujuan masing-masing *server*. Setelah semua *node* dibebani dengan *trafik* yang sudah ditentukan di atas, langkah selanjutnya yaitu mengatur waktu pengiriman tiap *trafik*. Pada simulasi ini, pengiriman dibagi menjadi 2, yaitu pengiriman *trafik* per *node* ke *server* dan pengiriman *trafik* seluruh *node* secara bersamaan ke *server*. Setelah pengaturan waktu *trafik* selesai, selanjutnya mengecek apakah pada saat *running script* terdapat *error* atau tidak, jika terdapat *error* maka akan dilakukan *tracing* ulang, jika tidak terdapat *error*, maka simulasi dilanjutkan ke pengujian. Berikut ini gambar 3.14 proses pembuatan *node* pada *software* NS2



**Gambar 3.14** Flowchart perancangan protokol *e-ticketing*

berikut ini gambar 3.15 yang merupakan gambaran dari *node* beserta *link* yang telah dibuat di *software* NS2



**Gambar 3.15** Hasil rancangan *node* yang sudah jadi

*Node* ini menggambarkan halte yang semua terhubung dengan jaringan fiber optik. Perhitungan jarak pada NS2 ini mengacu pada rumus (4) yang hasilnya terlampir pada lampiran.

Setelah implementasi sistem dan implementasi jaringan selesai dilakukan, selanjutnya dibuat skenario untuk pengujian hasil implementasi tersebut.

### **3.5 Skenario Pengujian Sistem *E-ticketing***

Setelah proses implementasi selesai, langkah selanjutnya yaitu menentukan skenario yang akan digunakan untuk pengujian sistem yang telah dirancang. Skenario tersebut meliputi skenario pengujian sistem dan skenario pengujian jaringan. Pada pengujian sistem, dibuat skenario pengujian fungsional, pengujian keamanan dan pengujian monitoring sistem. Pada pengujian jaringan, dibuat skenario pengujian *throughput*, *packet loss*, *delay* dan *jitter*.

#### **3.5.1 Skenario Pengujian Sistem**

Pada bagian ini akan dijabarkan mengenai skenario pengujian sistem *e-ticketing* yang telah dibuat, pengujian tersebut meliputi :

1. Pengujian fungsional sistem  
Pengujian ini bertujuan untuk mengetahui apakah sistem telah sesuai dengan rancangan atau belum. Pengujian ini dilakukan dengan cara melakukan proses *tap-in* pada website yang telah dibuat. Proses ini dilakukan untuk mengetahui apakah saldo dapat terpotong atau tidak.
2. Pengujian keamanan sistem  
Pengujian ini dilakukan untuk mengetahui apakah sistem yang telah dirancang aman untuk digunakan atau tidak. Pengujian ini dilakukan dengan cara, jika saldo dari kartu sudah habis atau tidak mencukupi maka sistem akan menolak transaksi tersebut.
3. Pengujian monitoring sistem  
Pengujian ini dilakukan untuk mengetahui apakah script maps yang dibuat sudah bisa menampilkan maps kota surabaya dengan titik yang menunjukkan posisi kendaraan saat ini.

### **3.5.2 Skenario Pengujian Jaringan**

Pada pengujian ini, akan dilihat QoS jaringan yang telah dirancang. QoS jaringan tersebut meliputi *throughput*, *packet loss*, *delay* dan *jitter*. Langkah-langkah pengujian yang dilakukan yaitu mengatur *trafik* atau besar file pada *tcl* dan berikan perintah *start* agar *trafik* tersebut terkirim ke *node* tujuan. Pada *script* yang dijalankan kan tersebut akan menghasilkan file baru, yaitu file *out.tr* dan file *out.nam*. File *out.tr* akan dianalisa sesuai dengan analisa jaringan QoS yaitu analisa *throughput*, *packet loss*, *delay* dan *jitter*. Analisa dilakukan dengan bantuan software pendukung, yaitu *trace graph*. Setelah didapatkan hasil dari software tersebut, langkah selanjutnya yaitu mencatat hasil kemudian diolah dengan *excel*. Dibuat grafik agar data yang dihasilkan dapat dibaca dan dipahami.



## BAB IV

### PENGUJIAN DAN ANALISIS

Setelah melakukan perancangan dan implementasi sistem *e-ticketing* dan jaringan *e-ticketing*, langkah selanjutnya yaitu melakukan pengujian perancangan tersebut. Pengujian dibagi menjadi 2 bagian, bagian pertama yaitu pengujian sistem *e-ticketing*, bagian kedua yaitu pengujian jaringan *e-ticketing*. Pengujian sistem *e-ticketing* meliputi pengujian fungsional sistem, pengujian keamanan sistem dan pengujian monitoring sistem. Pengujian jaringan *e-ticketing* meliputi pengujian *throughput*, *packet loss*, *delay* dan *jitter*.

#### 4.1 Pengujian Sistem *E-ticketing*

Pengujian ini dilakukan dengan cara menjalankan aplikasi *server* Xampp pada jaringan *localhost*. Pengujian tersebut berupa pengujian fungsional sistem, pengujian keamanan sistem dan pengujian monitoring sistem. Setelah semua pengujian dilakukan, selanjutnya akan dilakukan analisis hasil pengujian tersebut. Berikut ini penjelasan secara lengkap mengenai pengujian dan analisis yang akan dilakukan

##### 4.1.1 Pengujian Fungsional Sistem

Sesuai dengan skenario pengujian pada Bab 3, pada pengujian ini akan dilakukan pengujian fungsional sistem. Fungsi sistem yang diuji adalah fungsi *tap-in*, *tap-out* dan fungsi pemotongan saldo. Seperti dijelaskan pada bab perancangan sistem, pertama kali yang dilakukan yaitu masuk ke halaman ‘Update Live’. Tampilan pada gambar 4.1.

Halaman Utama | Daftar Tiket | Update Live | Rekap Akhir

Daftar kendaraan yang sedang beroperasi

No.	Kendaraan	Latitude	Longitude	Total User	Waktu	Posisi
1	monorail	-7.292082	112.722854	1	2016-01-07 17:03:19	monorail
2	tram	-7.236493	112.737597	1	2016-01-07 17:03:06	monorail

1

Proses Tap-in  
Kartu : 500  
Kendaraan : - Pilih Kendaraan -  
halte naik :  
Submit

Proses Tap-out  
Kartu : - Pilih Kartu -  
halte turun :  
Submit

**Gambar 4.1** Tampilan halaman ‘Update Live’

Selanjutnya masuk ke bagian *tap-in*, masukan id kartu, masukkan kendaraan yang akan digunakan dan halte sewaktu di lakukan proses tersebut. Pada simulasi ini, dilakukan proses *tap-in* menggunakan Id kartu 1001, kendaraan monorail dan posisi halte di bundaran ITS. Setelah itu klik *submit*, tampilan nya pada gambar 4.2.

**Gambar 4.2** Tampilan bagian *tap-in*

Kartu yang digunakan adalah kartu dengan id 1001. Kartu ini mempunyai saldo sebesar 5000. Kartu dengan id 1001 tersebut masuk ke dalam range kartu yang mendapat diskon sebesar 50%. Pada saat tombol *submit* di tekan, maka akan muncul halaman hasil *tap-in*, pada halaman ini terdapat beberapa informasi yang ditampilkan ke user, yaitu informasi harga tiket dan sisa saldo. Berikut ini gambar 4.3 tampilan hasil *tap-in*.

id = 1001  
 saldo awal = 5000  
 harga tiket = 2500  
 saldo akhir = 2500

**Gambar 4.3** Tampilan hasil *tap-in*

Setelah keluar hasil dari *tap-in* selanjutnya *refresh* pada halaman ‘Update Live’, pada halaman tersebut, user yang tadinya bernilai 1 setelah dilakukan proses *tap-in* maka user tambah 1 menjadi 2. Berikut ini gambar 4.4 yang merupakan tampilan dari halaman update live

#### Daftar kendaraan yang sedang beroperasi

No.	Kendaraan	Latitude	Longitude	Total User	Waktu	Posisi
1	monorail	-7.277030	112.803186	2	2016-01-07 17:11:13	<a href="#">monitor</a>
2	tram	-7.236493	112.737597	1	2016-01-07 17:03:06	<a href="#">monitor</a>

**Gambar 4.4** Tampilan halaman ‘Update Live’ setelah *refresh*

Setelah proses *tap-in* berhasil dilakukan, selanjutnya di uji untuk proses *tap-out*. Pada proses ini, id tiket yang sudah melakukan proses *tap-in* akan terlihat pada *combo box* bagian *tap-out*. Pada pengujian ini,

dilakukan proses *tap-out* untuk id tiket 1001 pada halte joyoboyo seperti pada gambar 4.5.

**Gambar 4.5** Tampilan bagian *tap-out*.

Setelah itu klik *submit*, maka akan tampil pesan yaitu “selamat anda berhasil melakukan *tap-out*”. Setelah itu, kembali ke halaman ‘Update Live’ kemudian *refresh* halaman tersebut. Total user yang tadinya berjumlah 2 orang, sekarang berubah menjadi 1 orang, seperti pada gambar 4.6.

#### Daftar kendaraan yang sedang beroperasi

No.	Kendaraan	Latitude	Longitude	Total User	Waktu	Posisi
1	monorail	-7.298378	112.737783	1	2016-01-07 17:26:30	<a href="#">monitor</a>
2	tram	-7.236493	112.737597	1	2016-01-07 17:03:06	<a href="#">monitor</a>

**Gambar 4.6** Tampilan halaman ‘Update Live’ setelah proses *tap-out*

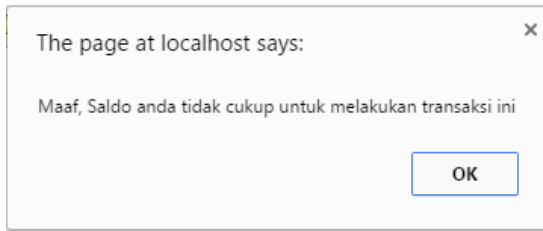
Setelah dilakukan pengujian proses *tap-in* dan proses *tap-out*, selanjutnya dilakukan pengujian keamanan sistem.

#### 4.1.2 Pengujian Keamanan Sistem

Pada pengujian ini akan dilakukan pengecekan keamanan sistem dengan cara jika saldo user tidak mencukupi untuk melakukan transaksi maka transaksi tersebut tidak bisa dilakukan. Pada tabel tiket, id 1002 mempunyai saldo sebesar 2000. Saldo tersebut tidak mencukupi untuk melakukan transaksi, sehingga sistem mengeluarkan alert bahwasannya saldo tidak mencukupi untuk melakukan transaksi. Berikut ini gambar 4.7 dan gambar 4.8 yang merupakan tampilan saldo tiket dan alert sistem.

No.	ID RFID	Saldo	5000	10000
1	500	17500	<a href="#">Tambah</a>	<a href="#">Tambah</a>
2	1001	5000	<a href="#">Tambah</a>	<a href="#">Tambah</a>
3	1002	2000	<a href="#">Tambah</a>	<a href="#">Tambah</a>
4	1003	5000	<a href="#">Tambah</a>	<a href="#">Tambah</a>
5	2001	6250	<a href="#">Tambah</a>	<a href="#">Tambah</a>

**Gambar 4.7** Tiket 1002 dengan saldo sebesar 2000



**Gambar 4.8** Alert sistem

#### 4.1.3 Pengujian Monitoring Sistem

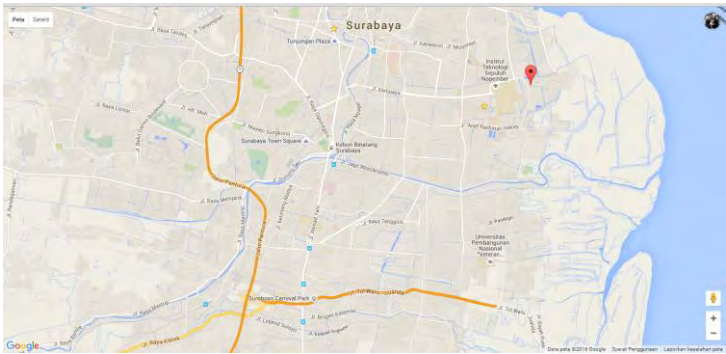
Pengujian ini dilakukan untuk mengetahui script map hasil rancangan dapat berfungsi atau tidak. Pada perancangan sistem, dibuat halte mengikuti jalur monorail dan jalur tram. Halte tersebut berjumlah 50 titik dengan masing-masing longitude dan latitude terdapat di database. Tabel 4.1 adalah daftar halte berikut koordinat nya. Pada nomor 1 – 23 adalah halte pada jalur monorail dan nomor 24 – 50 adalah halte pada jalur tram.

**Tabel 4.1** Halte dan koordinat halte

No	Halte	<i>Latitude</i>	<i>Longitude</i>
1	Kejawen	-7.27703	112.803186
2	Mulyosari	-7.274667	112.797886
3	ITS	-7.278818	112.790312
4	GOR Kertajawa Indah	-7.280499	112.781085
5	Dharmahusada Indah Timur	-7.275157	112.781664
6	Unair Kmapus C	-7.270048	112.782072
7	Darmahusada	-7.266281	112.766322
8	RS Dr Sutomo	-7.265827	112.758517
9	Stasiun Gubeng	-7.265657	112.75165
10	Jl Raya Gubeng	-7.272411	112.748319
11	Irian Barat	-7.277158	112.744298
12	Bung Tomo	-7.288942	112.744762
13	Ngagel	-7.296313	112.742289
14	Wonokromo	-7.300719	112.739629
15	Joyoboyo	-7.298378	112.737783
16	Adityawarman	-7.293546	112.730595

No	Halte	<i>Latitude</i>	<i>Longitude</i>
17	Pakis	-7.292082	112.722854
18	Dukuh Kupang	-7.290609	112.713944
19	Bundaran Satelit	-7.287523	112.704459
20	HR Muhammad	-7.285118	112.695404
21	Simpang Darmo Permai	-7.2815	112.684697
22	Lontar	-7.287523	112.678753
23	Unesa	-7.29874	112.675577
24	Lidah Kulon	-7.300095	112.662354
25	Joyoboyo Trem	-7.298569	112.737869
26	Bonbin	-7.29559	112.739264
27	Taman Bungkul	-7.291567	112.738985
28	Bintoro	-7.286352	112.739693
29	Pandegiling	-7.279775	112.740873
30	Panglima Sudirman	-7.274224	112.741973
31	Kombepol M Duryat	-7.269286	112.741533
32	Tegalsari	-7.265274	112.741018
33	Embong Malang	-7.261023	112.738963
34	Kedungdoro	-7.258895	112.733964
35	Pasar Blauran	-7.255078	112.734162
36	Bubutan	-7.2507	112.735766
37	Pasar Turi	-7.246518	112.737003
38	Kemayoran	-7.243239	112.734884
39	Indrapura	-7.237673	112.731037
40	Rajawali	-7.234906	112.730742
41	Jembatan Merah	-7.236493	112.737597
42	Veteran	-7.242542	112.738006
43	Tugu Pahlawan	-7.246507	112.73866
44	Baliwerti	-7.252513	112.736791
45	Siola	-7.255452	112.736911
46	Genteng	-7.258547	112.738411
47	Tunjungan	-7.261082	112.739937
48	Gub. Suryo	-7.264252	112.744955
49	Bambu Runcing	-7.267846	112.744561
50	Sonokembang	-7.271853	112.742887

Pada proses *tap in*, diisi halte pada saat naik kendaraan. Setelah proses *tap-in* berhasil, maka data pada halaman ‘Update Live’ harus di *refresh*, sehingga data koordinat berubah menjadi koordinat sewaktu proses tap-in dilakukan. Untuk mengetahui posisi dimana kendaraan tersebut, maka klik hyperlink ‘monitor’, maka pada browser akan membuka tab baru yang berisikan halaman map Kota Surabaya beserta titik yang menandakan posisi kendaraan. seperti pada gambar 4.9



**Gambar 4.9** Posisi kendaraan pada peta Kota Surabaya

**4.1.4 Analisis Hasil Pengujian**

Pengujian telah dilakukan, selanjutnya akan dianalisis hasil dari pengujian tersebut. Analisis tersebut meliputi apakah rancangan sistem *e-ticketing* yang telah dibuat, dapat berjalan sesuai perancangan atau tidak. Berikut ini tabel 4.2 yang merupakan hasil pengujian dan analisis hasil pengujian

**Tabel 4.2** Hasil simulasi

No	Pengujian	Hasil yang di dapat	Analisis
1	Fungsional	<ul style="list-style-type: none"><li>• Berhasil melakukan proses <i>tap-in</i> dan proses <i>tap-out</i></li><li>• Berhasil melakukan pemotongan saldo sesuai dengan kriteria tarif yang telah dirancang</li></ul>	Sistem telah sesuai dengan perancangan

No	Pengujian	Hasil yang di dapat	Analisis
2	Keamanan sistem	<ul style="list-style-type: none"> <li>Sistem berhasil menolak transaksi saat saldo pada kartu tidak mencukupi</li> </ul>	Sistem telah sesuai dengan perancangan
3	Monitoring sistem	<ul style="list-style-type: none"> <li>Berhasil melakukan <i>pin-point</i> titik koordinat pada google maps</li> </ul>	Sistem telah sesuai dengan perancangan

## 4.2 Pengujian Jaringan *E-ticketing*

Pengujian ini dilakukan untuk mengetahui kualitas jaringan yang sudah dibuat. Kualitas jaringan diukur mengikuti standart *Quality of Service* (QoS), yang diantara nya adalah *throughput*, *packet loss*, *delay* dan *jitter*.

Pada perancangan *node* di NS2, dibuat jaringan dengan kondisi *duplex-link* dengan *bandwith* 100 Mbps sesuai dengan kapasitas optik yang dipakai di jaringan. Pengujian dilakukan dengan cara membebani jaringan yang sudah di rancang dengan beban paket sebesar 100 kB. Beban paket 100 kB didapat dari perkiraan jumlah penumpang pada setiap kendaraan. Dalam simulasi ini, penulis membuat perancangan dengan satu kartu/tiket membawa data sebesar 2 kB, dan jumlah penumpang yang ada didalam kendaraan diperkirakan berjumlah 50 orang. Sehingga total *tap-in* atau data yang masuk ke OBU sebesar 100 kB.

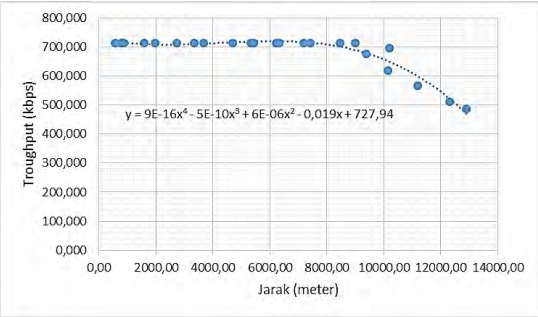
Pengujian dilakukan dalam 2 tahap, tahap pertama yaitu pengiriman data per *node*. Dalam tahap ini, data yang terdapat pada *node* dikirimkan satu persatu ke *server*. Pada tahap kedua, data yang terdapat pada setiap *node* dikirimkan secara bersamaan ke *server*. Berikut ini penjelasan secara lengkap mengenai pengujian dan analisis yang dilakukan

### 4.2.1 Pengujian dan analisis *throughput* jaringan

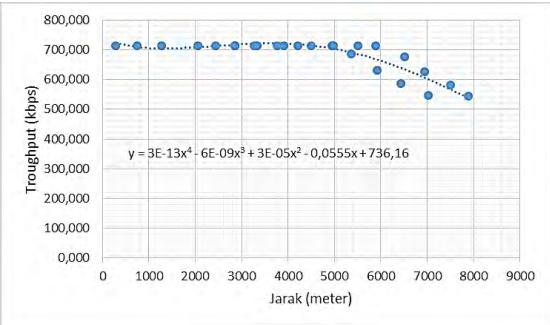
Pengujian ini dilakukan untuk mengetahui *throughput* jaringan atau bisa dikatakan untuk mengetahui *bandwidth* jaringan pada kondisi sebenarnya. Hasil dari implementasi di *software* NS2, kemudian di uji. Pengujian dilakukan dengan cara mengirimkan *trafik* dengan jenis FTP yang berjalan pada protokol TCP dengan paket sebesar 100 kB. Pengujian dilakukan selama 1 s. Pengujian dilakukan dalam 2 tahap, hasil dari pengujian tahap pertama dan kedua terdapat pada lampiran. Pada

simulasi tersebut, analisis dibedakan dalam 2 jalur, jalur monorail dan jalur tram.

Pada pengujian tahap pertama didapatkan hasil *throughput* paling besar terletak pada *node* yang mempunyai jarak paling dekat dengan *server* dan yang paling kecil terletak pada *node* yang mempunyai jarak paling jauh dengan *server*. Pada jalur monorail, jarak paling dekat dengan *server* adalah 568,81 meter dengan nilai *throughput* sebesar 714,432 kbps, dan jarak paling jauh dengan *server* adalah 12891,83 meter dengan nilai *throughput* sebesar 487,288 kbps. Pada jalur tram, jarak paling dekat dengan *server* adalah 285,6 meter dengan nilai *throughput* sebesar 714,440, dan jarak paling jauh dengan *server* adalah 7886,77 meter dengan nilai *throughput* sebesar 544,96. Berikut gambar 4.10 dan 4.11 grafik *polynomial throughput* terhadap jarak pada kedua jalur.



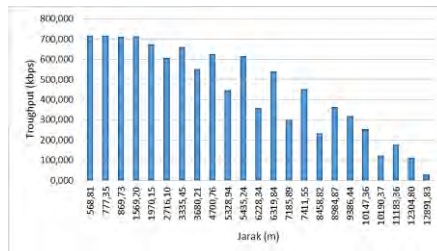
**Gambar 4.10** Grafik *polynomial throughput* terhadap jarak pada jalur *monorail* pengujian tahap pertama



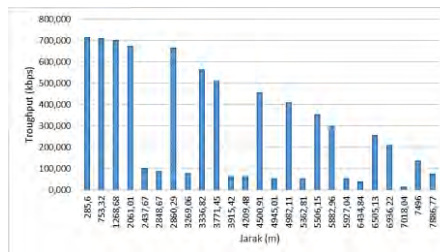
**Gambar 4.11** Grafik *polynomial throughput* terhadap jarak pada jalur *tram* pengujian tahap pertama



Pada pengujian tahap kedua, data pada setiap *node* dikirim ke *server* secara bersama-sama. Pengujian ini menghasilkan nilai *throughput* yang paling besar terletak pada *node* yang mempunyai jarak terpendek dengan *server*, sedangkan untuk *node* paling jauh dari *server*, menghasilkan nilai *throughput* yang kecil. Pada jalur monorail, jarak paling dekat dengan *server* yaitu 568,81 meter dengan nilai *throughput* sebesar 715,832 kbps, dan jarak paling jauh dengan *server* adalah 12891,83 meter dengan nilai *throughput* sebesar 29,344. Pada jalur tram, jarak paling dekat dengan *server* yaitu 285,6 meter dengan nilai *throughput* sebesar 715,416, dan jarak paling jauh dengan *server* adalah 2437,67 meter dengan nilai *throughput* sebesar 102.768 meter. Berikut gambar 4.12 dan gambar 4.13 perbandingan *throughput* terhadap jarak pada kedua jalur



**Gambar 4.12** Grafik perbandingan *throughput* terhadap jarak pada jalur monorail pengujian tahap kedua



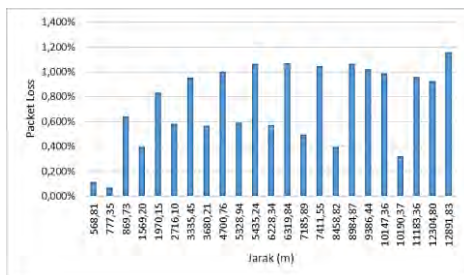
**Gambar 4.13** Grafik perbandingan *throughput* terhadap jarak pada jalur tram pengujian tahap kedua

Dengan hasil pengujian tersebut, dapat dianalisis jika jarak pengirim semakin jauh dari *server*, maka nilai *throughput* yang dihasilkan semakin kecil.

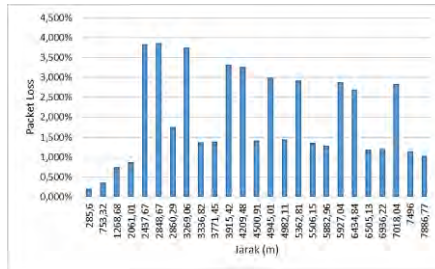
#### 4.2.2 Pengujian dan Analisis *packet loss* jaringan

Pengujian *packet loss* digunakan untuk mengetahui paket data yang hilang sewaktu dilakukan pengiriman dari *node* pengirim ke *server*. Pengujian ini dilakukan dengan cara yang sama dengan pengujian *throughput*, yaitu dengan mengirimkan paket data sebesar 100 kB selama 1s. Pengujian ini dilakukan dalam 2 tahap, tahap pertama yaitu pengiriman data pernode ke *server*. Hasil dari simulasi ini terdapat di tabel yang dilampirkan. Pengujian tahap pertama menghasilkan nilai *packet loss* sebesar 0%. Nilai tersebut didapat dari perbandingan jumlah paket yang hilang dibagi dengan jumlah paket yang dikirim selama pengujian berlangsung. Pengujian ini menghasilkan *packet loss* sebesar 0% dikarenakan jalur sebesar 100mb hanya dipergunakan untuk mengirimkan data yang kecil dan pengiriman tersebut dilakukan pernode tanpa ada gangguan dari *node* yang lain. Menurut standart ITU-T dan TIPHON *pacet loss* dengan nilai 0% memiliki kualitas sangat bagus.

Pada pengujian tahap kedua, menghasilkan nilai *packet loss* yang berbeda antara jalur monorail dan jalur tram. Pada jalur monorail, jarak terdekat dengan *server* yaitu 568,81 meter dengan nilai *packet loss* sebesar 0.115%. Dan jarak paling jauh dengan *server* adalah 12891,83 meter dengan nilai *packet loss* sebesar 1,159%. Pada jalur tram, jarak paling dekat dengan *server* yaitu 285,6 meter dengan nilai *packet loss* sebesar 0.205%. Dan jarak paling jauh dengan *server* adalah 7886,77 meter dengan nilai *packet loss* sebesar 1,033%. Berikut gambar 4.14 dan gambar 4.15 grafik perbandingan *packet loss* terhadap jarak pada kedua jalur



**Gambar 4.14** Grafik perbandingan *packet loss* terhadap jarak pada jalur monorail



**Gambar 4.15** Grafik perbandingan *packet loss* terhadap jarak pada jalur tram

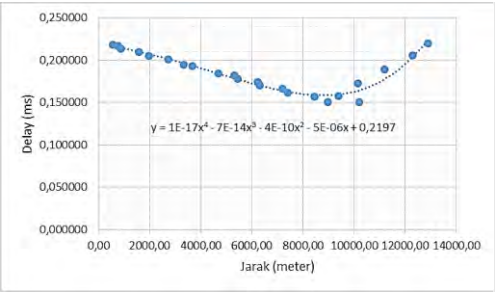
Pada gambar perbandingan diatas, dapat dianalisis bahwa pada jalur monorail, semaik jauh pengirim dari *server*, maka semakin besar *loss* yang terjadi. Tetapi pada jalur tram, hal tersebut tidak berlaku. Dapat dilihat jika jarak semakin jauh, *packet loss* yang terjadi semakin kecil, sedangkan *packet loss* terbesar terjadi pada jarak 2848,67 meter dengan nilai *packet loss* sebesar 3,863 %. Hal tersebut dapat terjadi dikarenakan terdapatnya penyempitan jalur atau penggabungan jalur. Pada simulasi tram, terdapat dua jalur yaitu jalur dari *node* 30 sampai dengan *node* 39 dan *node* 49 sampai dengan *node* 40. Kedua jalur tersebut terhubung ke nde 29. Sehingga sewaktu pengiriman data berlangsung, terjadi tabrakan yang menyebabkan data hilang dan menjadikan nilai *packet loss* semakin besar. Menurut standart ITU-T dan TIPHON *packet loss* dengan nilai 3,863% masuk dalam kualitas sedang/cukup.

#### 4.2.3 Pengujian dan Analisis *delay* jaringan

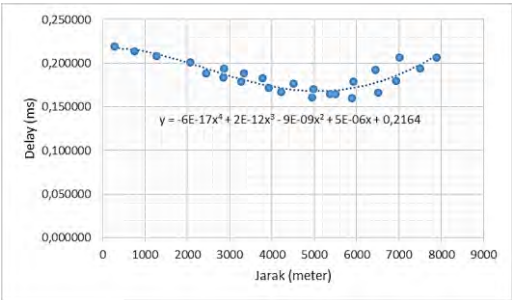
Pengujian *delay* dilakukan untuk mengetahui waktu tunda paket yang diakibatkan oleh proses transmisi dari satu *node* ke *node server*. Dalam pengujian *delay* jaringan akan dilakukan dalam dua tahap, tahap pertama pengiriman per*node* ke *server*, dan tahap kedua yaitu pengiriman semua *node* ke *server*. Pengujian ini dilakukan dengan cara yang sama dengan pengujian *throughput* dan pengujian *packet loss*, yaitu dengan mengirimkan paket data sebesar 100 kB selama 1 s.

Tabel hasil dari pengujian tahap pertama terdapat pada lampiran. Pada tahap pertama, simulasi *delay* menghasilkan grafik yang hampir sama pada kedua jalur. Pada jalur monorail, *delay* paling besar terjadi pada jarak 12891,83 meter dari *server* dengan nilai *delay* sebesar 0,220043 ms, sedangkan pada jalur tram *delay* terbesar terjadi pada jarak

285,60 meter dari *server* dengan nilai *delay* sebesar 0,219352 ms. Grafik yang ditunjukkan pada gambar hampir sama, tetapi mempunyai nilai yang berbeda. Pada jalur monorail, *delay* terbesar terjadi pada jarak terjauh, sedangkan pada jalur tram, *delay* terbesar terjadi pada jarak terdekat dari *server*. Normal *delay* yaitu jika jarak pengirim jauh dari *server*, maka *delay* lebih besar, tetapi pada jalur tram nilai *delay* menunjukkan jarak yang dekat mempunyai *delay* yang tinggi. Hal tersebut terjadi karena, pada 5 *node* terdekat dengan *server*, *node* tersebut merupakan penggabungan dari 2 jalur, sehingga data masuk ke jalur yang lebih sempit sehingga mengakibatkan terjadinya *delay* yang besar. Nilai *delay* terbesar pada pengujian tahap pertama yaitu sebesar 0,220043 ms, menurut standar ITU-T dan TIPHONE nilai tersebut memenuhi standar dengan kualitas baik di karenakan nilainya <150 ms. Berikut ini gambar 4.16 dan gambar 4.17 perbandingan antara *delay* terhadap jarak pada kedua jalur



**Gambar 4.16** Grafik *polynomial delay* terhadap jarak pada jalur monorail pengujian tahap pertama

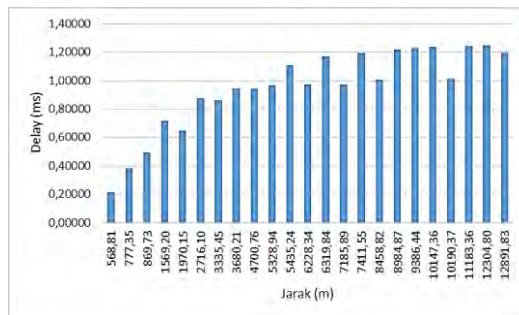


**Gambar 4.17** Grafik *polynomial delay* terhadap jarak pada jalur tram pengujian tahap pertama

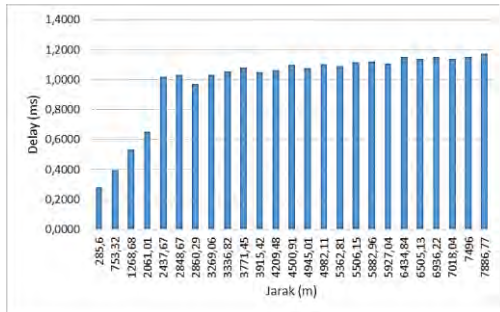
Pengujian tahap kedua yaitu mengirimkan data dari semua *node* ke *server*. Tabel hasil dari pengujian ini terdapat di lampiran. Analisis dari pengujian tahap kedua ini yaitu nilai *delay* pada kedua jalur tidak jauh berbeda. Pada jalur monorail didapat *delay* terbesar terjadi pada jarak 12304,8 meter dengan nilai *delay* sebesar 1,24434 ms, sedangkan pada jalur tram, nilai *delay* terbesar terdapat pada jarak 7886,77 meter dengan nilai *delay* sebesar 1,1736 ms. Pada pengujian tahap kedua ini, rata2 terjadi peningkatan *delay* diatas 1ms terjadi pada jarak diatas 5000 meter untuk jalur monorail dan diatas jarak 2400 meter untuk jalur tram.

Hal tersebut dapat berbeda dikarenakan pada jalur monorail, hanya terdapat satu jalur untuk terhubung ke *server*. *Server* berada ditengah-tengah jalur, *server* menghubungkan jalur timur ke barat. Sedangkan pada jalur tram, jalur ini menghubungkan dari utara ke selatan, tetapi pada *node* ke 5 terdapat pemisahan jalur, sehingga dapat dikatakan terdapat 2 jalur yang saling bertemu di *node* ke 5. Jarak *node* ke 5 dengan *server* yaitu 2400 meter, oleh karena itu, *delay* mulai meningkat setelah *node* ke 5.

Menurut standar ITU-T dan TIPHON, *delay* terbesar pada pengujian ini terletak pada jalur monorail dengan *delay* sebesar 1,24434 yang masuk pada kualitas baik/sangat bagus. Berikut ini gambar 4.18 dan gambar 4.19 grafik perbandingan *delay* terhadap jarak pada kedua jalur



**Gambar 4.18** Grafik perbandingan *delay* terhadap jarak pada jalur monorail pengujian tahap kedua

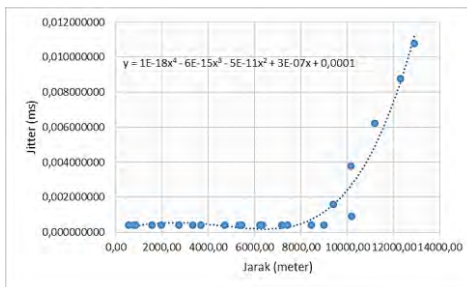


**Gambar 4.19** Grafik perbandingan *delay* terhadap jarak pada jalur tram pengujian tahap kedua

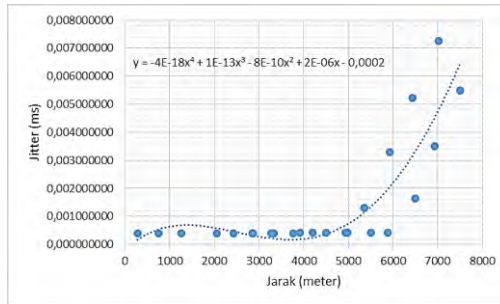
#### 4.2.4 Pengujian dan Analisis *jitter* jaringan

Pengujian *jitter* dilakukan untuk mengetahui variasi dari *delay* yang terjadi akibat adanya selisih waktu atau *interval* antara kedatangan paket pada penerima. Pengujian ini dilakukan dengan cara yang sama dengan simulasi sebelum nya, yaitu dengan mengirimkan paket data sebesar 100 kB selama 1 s

Pengujian ini dilakukan dalam dua tahap, yaitu data dikirim satu persatu ke *server* dan data dikirimkan seraca bersama ke *server*. Hasil dari pengujian tahap pertama, yaitu nilai terbesar pada jalur monorail terdapat pada jarak terjauh yaitu 12891,83 meter dengan nilai *jitter* sebesar 0,010792 ms, dan pada jalur tram nilai *jitter* terbesar terdapat pada jarak 7886,77 meter dengan nilai *jitter* sebesar 0,00731944 ms. Berikut ini gambar 4.20 dan gambar 4.21 perbandingan antara *jitter* dengan jarak pada kedua jalur.



**Gambar 4.20** Grafik *polynomial jitter* terhadap jarak pada jalur monorail pengujian tahap pertama



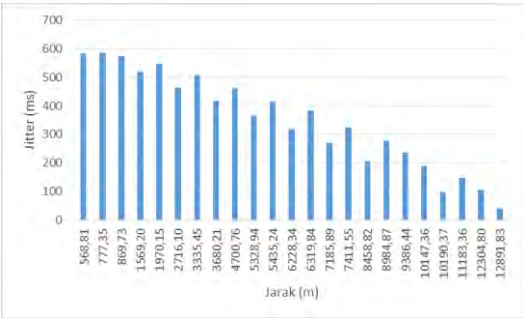
**Gambar 4.21** Grafik *polynomial jitter* terhadap jarak pada jalur tram pengujian tahap pertama

Pada kedua grafik diatas, nilai *jitter* mulai besar terjadi ketika jarak 9000 meter keatas pada jalur monorail dan untuk jalur tram peningkatan nilai *jitter* terjadi pada jarak 5000 meter keatas. Peningkatan tersebut dapat terjadi karena jarak yang cukup jauh menyebabkan semakin besarnya variasi dari *delay*. Tetapi untuk pengujian tahap pertama ini, nilai *jitter* yang terbesar masih memenuhi kualitas standar baik dari ITU-T dan TIPHON

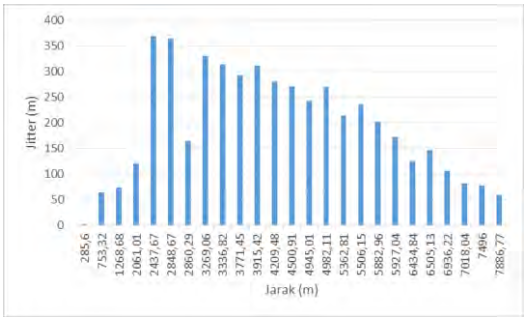
Pengujian tahap kedua, dilakukan dengan mengirimkan data dari seluruh *node* ke *server*. Pengiriman tersebut menghasilkan nilai yang berbeda-beda pada jalur tram dan jalur monorail. Pada jalur monorail, nilai *jitter* paling besar terletak pada *node* yang terdekat dengan *server* dengan nilai sebesar 583,739 ms, dan nilai yang paling kecil terletak pada *node* yang paling jauh dari *server* dengan nilai sebesar 29,344 ms. Nilai *jitter* berkebalikan dengan nilai *delay*, karena pada *delay* nilai terbesar terletak pada *node* terjauh dari *server*. Hal ini menunjukkan bahwa pada jalur monorail, dengan *node* terdekat dengan *server*, mempunyai variasi *delay* yang sangat besar. Variasi tersebut dikarenakan banyaknya data yang dikirim ke *server*, data dari *node* ujung sampai dengan *node* pangkal berkumpul pada *node* 13 dan 15 yang menyebabkan tingginya nilai *jitter*.

Pengujian pada jalur tram menghasilkan grafik yang tidak jauh berbeda dari jalur monorail. Pada jalur ini, *jitter* terbesar terletak pada *node* nomor ke-5 dengan nilai sebesar 369,399 ms, dan nilai *jitter* paling kecil terletak pada *node* ke-1 sebesar 0.0163428 ms. Jitter paling besar terletak pada *node* ke-5, hal itu disebabkan karena pada *node* ke-5 terjadi percabangan jalur, sehingga *node* tersebut menerima banyak data yang harus di transmisikan ke *server*, dan pada *node* ke-4 sampai dengan *node*

ke-1 nilai *jitter* semakin kecil, hal itu disebabkan karena pada *node-node* tersebut, data yang harusnya dikirim mengalami *loss* sehingga data yang dikirim menjadi sedikit. Berikut ini gambar 4.22 dan gambar 4.23 grafik dari nilai *jitter*



**Gambar 4.22** Grafik perbandingan *jitter* terhadap jarak pada jalur monorail pengujian tahap kedua



**Gambar 4.23** Grafik perbandingan *jitter* terhadap jarak pada jalur tram pengujian tahap kedua

Pada kedua grafik diatas, nilai *jitter* paling besar terdapat di jalur monorail dengan nilai sebesar 583,739 ms. Menurut standar TIPHONE nilai *jitter* tersebut termasuk ke dalam kualitas jelek karena >225 ms.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah dilakukan perancangan dan pengujian sistem *e-ticketing* dan jaringan *e-ticketing*, maka secara keseluruhan dapat diambil kesimpulan sebagai berikut :

1. Pengujian sistem *e-ticketing* mengenai pengujian fungsional, pengujian keamanan sistem, pengujian monitoring sistem telah berjalan sesuai rancangan. Sistem yang disimulasikan mampu memenuhi kebutuhan selama proses *tap-in* sampai dengan *tap-out*.
2. Pengujian jaringan *e-ticketing*, dilakukan dalam dua tahap. Tahap pertama yaitu pengujian pengiriman data dari setiap *node* ke *server*. Hasil dari pengujian ini diambil hasil yang paling besar, yaitu *throughput* sebesar 714,440 kbps, *packet loss* sebesar 0 % (*perfect*), *delay* sebesar 0.220043 ms (<150 ms *perfect*) dan *jitter* sebesar 0.010792 ms (0 – 75 ms *good*).
3. Pengujian jaringan *e-ticketing* tahap kedua yaitu pengiriman data dari *node* dikirim ke *server* secara bersamaan. Hasil dari pengujian ini diambil hasil paling besar, yaitu nilai *throughput* sebesar 715,832 kbps, *packet loss* sebesar 3,8628 % ( 3%- 15% *good*), *delay* sebesar 1,24434 ms (<150 ms *perfect*) dan *jitter* sebesar 583,739 ms (>225 ms *poor*).
4. Berdasarkan standar QoS, terdapat 1 pengujian yang bernilai dibawah standar, yaitu pengujian *jitter* yang terdapat pada pengujian tahap kedua jalur monorail dengan standar QoS >225 ms dengan kualitas *poor*.
5. Total *throughput* pada pengujian kedua sebesar 10292 kbps pada jalur monorail dan 7886 kbps pada jalur tram, sehingga untuk penghematan dapat menyewa *bandwidth* dibawah 15 Mbps pada masing-masing jalur.

## 5.2 Saran

Adapun hal – hal yang masih bisa dikembangkan dari protokol *e-ticketing* ini adalah :

1. Pengembangan untuk armada jenis *feeder* dan *trunk*
2. Pengembangan dalam sistem pentarifan, dilihat dari berbagai macam sudut pandang dan berbagai macam harga
3. Pengembangan sistem keamanan dengan lebih mempertimbangkan faktor kehidupan sosial masyarakat di Indonesia

## LAMPIRAN

### A. Lembar pengesahan proposal

**Jurusan Teknik Elektro**  
Fakultas Teknologi Industri - ITS

**TE141599 TUGAS AKHIR – 4 SKS**

15 SEP 2015

Nama Mahasiswa : Prasetyo Yulianto  
Nomor Pokok : 2213 106 042  
Bidang Studi : Telekomunikasi Multimedia  
Tugas Diberikan : Semester Gasal Th. 2015/2016  
Dosen Pembimbing : 1. Dr. Ir. Endroyono, DEA  
2. Ir. Gatot Kusrahardjo, MT

Judul Tugas Akhir : **Rancang Bangun Protokol E-Ticketing**  
(*Design of E-Ticketing Protocol*)

**Uraian Tugas Akhir :**  
Transportasi modern membutuhkan penerapan Teknologi Informasi dan Komunikasi (TIK), diantaranya adalah teknologi e-ticketing yang membutuhkan dukungan perangkat digital dan jaringan komunikasi. Agar pembayaran elektronik dapat berlangsung aman, lancar dan sesuai dengan konsep *management revenue*, maka protokol komunikasi harus dirancang dengan baik. Tugas akhir ini bertujuan merancang protokol untuk sistem e-ticketing mulai dari pembacaan data di tiket (RFID), penyimpanan data di *On Board Unit* (OBU), komunikasi dari OBU ke *access point*, hingga komunikasi internet ke *server*. Evaluasi kinerja dilakukan dengan simulasi menggunakan *software* NS-2, terkait standart *Quality of Service* (QoS) dengan kualitas baik, dengan memvariasikan jumlah node di Surabaya.

**Kata kunci :** E-Ticketing, NS2, QoS

Dosen Pembimbing 1, Dosen Pembimbing 2,


  
**Dr. Ir. Endroyono, DEA**  
NIP. 196504041991021001

  
**Ir. Gatot Kusrahardjo, MT,**  
NIP. 195904281986011001

Mengetahui,  
Jurusan Teknik Elektro FTI – ITS  
Ketua,

  
**Dr. Tri Arief Sardjono, ST., MT.**  
NIP. 197002121995121001

Menyetujui,  
Bidang Studi Telekomunikasi Multimedia  
Koordinator,

  
**Dr. Ir. Endroyono, DEA**  
NIP. 196504041991021001

*[ Halaman ini sengaja dikosongkan ]*

## B. Tabel hasil pengujian

1. Pengujian tahap pertama (data dikirim per node ke server) pada jalur monorail

Node Pengirim	packet loss	Delay (ms)	jitter (ms)	Throughput (kbps)	Jarak (m)
0	0%	0,220043	0,010792000	487,288	12891,83
1	0%	0,205963	0,008774540	512,216	12304,80
2	0%	0,189262	0,006231110	565,936	11183,36
3	0%	0,172980	0,003771140	619,872	10147,36
4	0%	0,158047	0,001581960	677,312	9386,44
5	0%	0,151014	0,000414258	713,816	8984,87
6	0%	0,161997	0,000411230	713,936	7411,55
7	0%	0,170587	0,000409102	714,032	6319,84
8	0%	0,177979	0,000407302	714,104	5435,24
9	0%	0,184771	0,000405569	714,176	4700,76
10	0%	0,194762	0,000403376	714,272	3335,45
11	0%	0,204757	0,000401846	714,344	1970,15
12	0%	0,213354	0,000400600	714,408	869,73
13	0%	0,217953	0,000400083	714,432	568,81
15	0%	0,216953	0,000400105	714,432	777,35
16	0%	0,209754	0,000400822	714,392	1569,20
17	0%	0,200958	0,000402251	714,320	2716,10
18	0%	0,193163	0,000403698	714,256	3680,21
19	0%	0,181774	0,000406065	714,152	5328,94
20	0%	0,174182	0,000407887	714,080	6228,34
21	0%	0,166390	0,000409759	714,000	7185,89
22	0%	0,156803	0,000412307	713,896	8458,82
23	0%	0,150789	0,000929608	696,552	10190,37

2. Pengujian tahap pertama (data dikirim per node ke server) pada jalur tram

Node Pengirim	packet loss	Delay (ms)	jitter (ms)	Throughput (kbps)	Jarak (m)
25	0%	0,219352	0,000400045	714,440	285,60
26	0%	0,213754	0,000400567	714,408	753,32
27	0%	0,208156	0,000401433	714,360	1268,68
28	0%	0,201160	0,000402686	714,304	2061,01
29	0%	0,193965	0,000404178	714,240	2860,29
30	0%	0,188372	0,000405682	714,176	3336,82
31	0%	0,182978	0,000407114	714,120	3771,45
32	0%	0,176186	0,000408824	714,048	4500,91
33	0%	0,170592	0,000410340	713,984	4982,11
34	0%	0,165003	0,000412280	713,904	5506,15
35	0%	0,160013	0,000414092	713,832	5882,96
36	0%	0,166317	0,001636180	675,768	6505,13
37	0%	0,179633	0,003501400	626,424	6936,22
38	0%	0,193580	0,005493660	581,096	7496,00
39	0%	0,206697	0,007319440	544,960	7886,77
40	0%	0,206356	0,007250480	546,224	7018,04
41	0%	0,192295	0,005235850	586,584	6434,84
42	0%	0,178604	0,003295460	631,520	5927,04
43	0%	0,164635	0,001299300	685,528	5362,81
44	0%	0,160612	0,000413975	713,840	4945,01
45	0%	0,167400	0,000411806	713,928	4209,48
46	0%	0,171991	0,000410122	713,992	3915,42
47	0%	0,178584	0,000408501	714,064	3269,06
48	0%	0,183778	0,000407035	714,120	2848,67
49	0%	0,188771	0,000405599	714,184	2437,67

3. Pengujian tahap kedua (data dikirim secara bersamaan ke server) pada jalur monorail

Node Pengirim	Packet Loss	Delay (ms)	Jitter (ms)	Throughput (kbps)	jarak (m)
0	1,159%	1,19429	40,0151	29,344	12891,83
1	0,925%	1,24434	106,178	111,408	12304,80
2	0,959%	1,24375	148,453	176,040	11183,36
3	0,988%	1,23532	189,872	254,904	10147,36
4	1,019%	1,22543	236,368	317,696	9386,44
5	1,064%	1,21641	277,093	362,888	8984,87
6	1,047%	1,19239	324,241	451,568	7411,55
7	1,070%	1,17078	381,744	539,824	6319,84
8	1,064%	1,10868	414,499	613,680	5435,24
9	1,004%	0,94098	461,122	625,704	4700,76
10	0,954%	0,85960	507,139	661,176	3335,45
11	0,833%	0,64602	545,88	672,296	1970,15
12	0,642%	0,49140	572,504	712,208	869,73
13	0,115%	0,21526	583,739	715,832	568,81
15	0,069%	0,38187	585,12	714,984	777,35
16	0,397%	0,71495	520,652	712,624	1569,20
17	0,582%	0,87704	463,553	607,656	2716,10
18	0,569%	0,94307	416,125	550,872	3680,21
19	0,590%	0,96697	365,755	447,520	5328,94
20	0,572%	0,97124	320,359	358,776	6228,34
21	0,493%	0,97164	271,229	298,560	7185,89
22	0,399%	1,00280	206,823	233,936	8458,82
23	0,323%	1,01019	96,4504	123,376	10190,37

4. Pengujian tahap kedua (data dikirim secara bersamaan ke server) pada jalur tram

Node Pengirim	Packet Loss	Delay (ms)	Jitter (ms)	Throughput (kbps)	jarak (m)
25	0,205%	0,2820	0,016453	715,416	285,6
26	0,356%	0,3942	63,6979	711,264	753,32
27	0,745%	0,5336	73,3713	704,240	1268,68
28	0,871%	0,6541	120,416	675,640	2061,01
29	1,756%	0,9719	163,755	667,072	2860,29
30	1,376%	1,0526	314,183	564,240	3336,82
31	1,385%	1,0810	292,735	512,168	3771,45
32	1,410%	1,0975	271,395	458,552	4500,91
33	1,441%	1,1017	269,681	411,576	4982,11
34	1,364%	1,1154	235,791	354,504	5506,15
35	1,292%	1,1200	201,174	300,192	5882,96
36	1,185%	1,1398	146,519	258,105	6505,13
37	1,200%	1,1528	105,656	210,728	6936,22
38	1,148%	1,1500	77,3727	137,072	7496
39	1,033%	1,1736	58,0148	76,680	7886,77
40	2,843%	1,1376	80,7899	15,584	7018,04
41	2,697%	1,1531	125,377	40,704	6434,84
42	2,873%	1,1061	171,875	55,704	5927,04
43	2,926%	1,0904	214,56	55,728	5362,81
44	2,993%	1,0768	242,641	55,752	4945,01
45	3,262%	1,0635	280,657	65,016	4209,48
46	3,327%	1,0494	311,125	65,088	3915,42
47	3,748%	1,0344	330,395	79,712	3269,06
48	3,863%	1,0340	363,826	88,800	2848,67
49	3,826%	1,0193	369,399	102,768	2437,67



**C. Tabel hasil pengujian**

No	Node A	Node B	Jarak (m)	Delay Link (ms)
1	0	1	587,03	0.0028786129
2	1	2	1121,44	0.0054991937
3	2	3	1036	0.0050802225
4	3	4	760,92	0.0037313155
5	4	5	401,57	0.0019691746
6	5	6	1573,32	0.0077150730
7	6	7	1091,71	0.0053534070
8	7	8	884,6	0.0043378038
9	8	9	734,48	0.0036016620
10	9	10	1365,31	0.0066950565
11	10	11	1365,3	0.0066950074
12	11	12	1100,42	0.0053961181
13	12	13	300,92	0.0014756183
14	13	14	568,81	0.0027892677
15	14	15	777,35	0.0038118831
16	15	16	791,85	0.0038829866
17	16	17	1146,9	0.0056240416
18	17	18	964,11	0.0047276962
19	18	19	1648,73	0.0080848602
20	19	20	899,4	0.0044103784
21	20	21	957,55	0.0046955280
22	21	22	1272,93	0.0062420536
23	22	23	1731,55	0.0084909838
24	24	25	285,6	0.0014004938
25	25	26	467,72	0.0022935537
26	26	27	515,36	0.0025271655
27	27	28	792,33	0.0038853404
28	28	29	799,28	0.0039194210
29	29	30	476,53	0.0023367552
30	30	31	434,63	0.0021312906
31	31	32	729,46	0.0035770454
32	32	33	481,2	0.0023596554
33	33	34	524,04	0.0025697295
34	34	35	376,81	0.0018477593
35	35	36	622,17	0.0030509286

No	<i>Node A</i>	<i>Node B</i>	Jarak (m)	<i>Delay Link</i> (ms)
36	36	37	431,09	0.0021139315
37	37	38	559,78	0.0027449874
38	38	39	390,77	0.0019162148
39	39	40	1033,32	0.0050670806
40	40	41	583,20	0.0028598318
41	41	42	507,8	0.0024900936
42	42	43	564,23	0.0027668088
43	43	44	417,8	0.0020487615
44	44	45	735,53	0.0036068108
45	45	46	294,06	0.0014419790
46	46	47	646,36	0.0031695488
47	47	48	420,39	0.0020614621
48	48	49	411	0.0020154164
49	49	29	376,66	0.0018470237

## Listing Program

### -Program cek.tcl (pengiriman per node ke server)

```
#Create a simulator object  
set ns [new Simulator]
```

```
#Perbedaan aliran data (NAM)
```

```
$ns color 0 blue  
$ns color 1 blue  
$ns color 2 blue  
$ns color 3 blue  
$ns color 4 blue  
$ns color 5 blue  
$ns color 6 blue  
$ns color 7 blue  
$ns color 8 blue  
$ns color 9 blue  
$ns color 10 blue  
$ns color 11 blue  
$ns color 12 blue  
$ns color 13 blue  
$ns color 14 blue  
$ns color 15 blue  
$ns color 16 blue  
$ns color 17 blue  
$ns color 18 blue  
$ns color 19 blue  
$ns color 20 blue  
$ns color 21 blue  
$ns color 22 blue  
$ns color 23 blue  
$ns color 24 blue  
$ns color 25 blue  
$ns color 26 blue  
$ns color 27 blue  
$ns color 28 blue  
$ns color 29 blue  
$ns color 30 blue
```

```
$ns color 31 blue  
$ns color 32 blue  
$ns color 33 blue  
$ns color 34 blue  
$ns color 35 blue  
$ns color 36 blue  
$ns color 37 blue  
$ns color 38 blue  
$ns color 39 blue  
$ns color 40 blue  
$ns color 41 blue  
$ns color 42 blue  
$ns color 43 blue  
$ns color 44 blue  
$ns color 45 blue  
$ns color 46 blue  
$ns color 47 blue  
$ns color 48 blue  
$ns color 49 blue
```

```
#Membuka Trace file  
set file1 [open outtrace-baru.tr  
w]  
set winfile [open WinFile w]  
$ns trace-all $file1
```

```
#Membuka NAM trace file  
set file2 [open outtrace-  
baru.nam w]  
$ns namtrace-all $file2
```

```
#Eksekusi terakhir  
proc finish {} {  
    global ns file1 file2  
    $ns flush-trace  
    close $file1  
    close $file2  
    #exec nam outtrace-  
baru.nam &
```

```

        exit 0
    }

#Membuat node
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
set n9 [$ns node]
set n10 [$ns node]
set n11 [$ns node]
set n12 [$ns node]
set n13 [$ns node]
set n14 [$ns node]
set n15 [$ns node]
set n16 [$ns node]
set n17 [$ns node]
set n18 [$ns node]
set n19 [$ns node]
set n20 [$ns node]
set n21 [$ns node]
set n22 [$ns node]
set n23 [$ns node]
set n24 [$ns node]
set n25 [$ns node]
set n26 [$ns node]
set n27 [$ns node]
set n28 [$ns node]
set n29 [$ns node]
set n30 [$ns node]
set n31 [$ns node]
set n32 [$ns node]
set n33 [$ns node]

set n34 [$ns node]
set n35 [$ns node]
set n36 [$ns node]
set n37 [$ns node]
set n38 [$ns node]
set n39 [$ns node]
set n40 [$ns node]
set n41 [$ns node]
set n42 [$ns node]
set n43 [$ns node]
set n44 [$ns node]
set n45 [$ns node]
set n46 [$ns node]
set n47 [$ns node]
set n48 [$ns node]
set n49 [$ns node]

$ns14 shape box
$ns14 color red
$ns24 shape box
$ns24 color red

#membuat label
$ns0 label "M-1"
$ns1 label "M-2"
$ns2 label "M-3"
$ns3 label "M-4"
$ns4 label "M-5"
$ns5 label "M-6"
$ns6 label "M-7"
$ns7 label "M-8"
$ns8 label "M-9"
$ns9 label "M-10"
$ns10 label "M-11"
$ns11 label "M-12"
$ns12 label "M-13"
$ns13 label "M-14"
$ns14 label "M-15"
$ns15 label "M-16"

```

\$n16 label "M-17"	\$ns duplex-link \$n0 \$n1
\$n17 label "M-18"	100Mb 0.0028786129ms
\$n18 label "M-19"	DropTail
\$n19 label "M-20"	\$ns duplex-link \$n1 \$n2
\$n20 label "M-21"	100Mb 0.0054991937ms
\$n21 label "M-22"	DropTail
\$n22 label "M-23"	\$ns duplex-link \$n2 \$n3
\$n23 label "M-24"	100Mb 0.0050802225ms
\$n24 label "T-1"	DropTail
\$n25 label "T-2"	\$ns duplex-link \$n3 \$n4
\$n26 label "T-3"	100Mb 0.0037313155ms
\$n27 label "T-4"	DropTail
\$n28 label "T-5"	\$ns duplex-link \$n4 \$n5
\$n29 label "T-6"	100Mb 0.0019691746ms
\$n30 label "T-7"	DropTail
\$n31 label "T-8"	\$ns duplex-link \$n5 \$n6
\$n32 label "T-9"	100Mb 0.0077150730ms
\$n33 label "T-10"	DropTail
\$n34 label "T-11"	\$ns duplex-link \$n6 \$n7
\$n35 label "T-12"	100Mb 0.0053534070ms
\$n36 label "T-13"	DropTail
\$n37 label "T-14"	\$ns duplex-link \$n7 \$n8
\$n38 label "T-15"	100Mb 0.0043378038ms
\$n39 label "T-16"	DropTail
\$n40 label "T-17"	\$ns duplex-link \$n8 \$n9
\$n41 label "T-18"	100Mb 0.0036016620ms
\$n42 label "T-19"	DropTail
\$n43 label "T-20"	\$ns duplex-link \$n9 \$n10
\$n44 label "T-21"	100Mb 0.0066950565ms
\$n45 label "T-22"	DropTail
\$n46 label "T-23"	\$ns duplex-link \$n10 \$n11
\$n47 label "T-24"	100Mb 0.0066950074ms
\$n48 label "T-25"	DropTail
\$n49 label "T-26"	\$ns duplex-link \$n11 \$n12
	100Mb 0.0053961181ms
	DropTail
	\$ns duplex-link \$n12 \$n13
	100Mb 0.0014756183ms
	DropTail

#Membuat link diantara *node*

\$ns duplex-link \$n13 \$n14  
 100Mb 0.0027892677ms  
 DropTail  
 \$ns duplex-link \$n14 \$n15  
 100Mb 0.0038118831ms  
 DropTail  
 \$ns duplex-link \$n15 \$n16  
 100Mb 0.0038829866ms  
 DropTail  
 \$ns duplex-link \$n16 \$n17  
 100Mb 0.0056240416ms  
 DropTail  
 \$ns duplex-link \$n17 \$n18  
 100Mb 0.0047276962ms  
 DropTail  
 \$ns duplex-link \$n18 \$n19  
 100Mb 0.0080848602ms  
 DropTail  
 \$ns duplex-link \$n19 \$n20  
 100Mb 0.0044103784ms  
 DropTail  
 \$ns duplex-link \$n20 \$n21  
 100Mb 0.0046955280ms  
 DropTail  
 \$ns duplex-link \$n21 \$n22  
 100Mb 0.0062420536ms  
 DropTail  
 \$ns duplex-link \$n22 \$n23  
 100Mb 0.0084909838ms  
 DropTail  
 \$ns duplex-link \$n24 \$n25  
 100Mb 0.0014004938ms  
 DropTail  
 \$ns duplex-link \$n25 \$n26  
 100Mb 0.0022935537ms  
 DropTail  
 \$ns duplex-link \$n26 \$n27  
 100Mb 0.0025271655ms  
 DropTail

\$ns duplex-link \$n27 \$n28  
 100Mb 0.0038853404ms  
 DropTail  
 \$ns duplex-link \$n28 \$n29  
 100Mb 0.0039194210ms  
 DropTail  
 \$ns duplex-link \$n29 \$n30  
 100Mb 0.0023367552ms  
 DropTail  
 \$ns duplex-link \$n30 \$n31  
 100Mb 0.0021312906ms  
 DropTail  
 \$ns duplex-link \$n31 \$n32  
 100Mb 0.0035770454ms  
 DropTail  
 \$ns duplex-link \$n32 \$n33  
 100Mb 0.0023596554ms  
 DropTail  
 \$ns duplex-link \$n33 \$n34  
 100Mb 0.0025697295ms  
 DropTail  
 \$ns duplex-link \$n34 \$n35  
 100Mb 0.0018477593ms  
 DropTail  
 \$ns duplex-link \$n35 \$n36  
 100Mb 0.0030509286ms  
 DropTail  
 \$ns duplex-link \$n36 \$n37  
 100Mb 0.0021139316ms  
 DropTail  
 \$ns duplex-link \$n37 \$n38  
 100Mb 0.0027449874ms  
 DropTail  
 \$ns duplex-link \$n38 \$n39  
 100Mb 0.0019162148ms  
 DropTail  
 \$ns duplex-link \$n39 \$n40  
 100Mb 0.0050670806ms  
 DropTail

\$ns duplex-link \$n40 \$n41  
 100Mb 0.0028598318ms  
 DropTail  
 \$ns duplex-link \$n41 \$n42  
 100Mb 0.0024900936ms  
 DropTail  
 \$ns duplex-link \$n42 \$n43  
 100Mb 0.0027668088ms  
 DropTail  
 \$ns duplex-link \$n43 \$n44  
 100Mb 0.0020487615ms  
 DropTail  
 \$ns duplex-link \$n44 \$n45  
 100Mb 0.0036068108ms  
 DropTail  
 \$ns duplex-link \$n45 \$n46  
 100Mb 0.0014419790ms  
 DropTail  
 \$ns duplex-link \$n46 \$n47  
 100Mb 0.0031695488ms  
 DropTail  
 \$ns duplex-link \$n47 \$n48  
 100Mb 0.0020614621ms  
 DropTail  
 \$ns duplex-link \$n48 \$n49  
 100Mb 0.0020154164ms  
 DropTail  
 \$ns duplex-link \$n49 \$n29  
 100Mb 0.0018470237ms  
 DropTail  
 \$ns duplex-link \$n24 \$n14  
 100Mb 0.0012307812ms  
 DropTail  
  
 \$ns duplex-link-op \$n0 \$n1  
 orient left  
 \$ns duplex-link-op \$n1 \$n2  
 orient left

\$ns duplex-link-op \$n2 \$n3  
 orient left  
 \$ns duplex-link-op \$n3 \$n4  
 orient left  
 \$ns duplex-link-op \$n4 \$n5  
 orient left  
 \$ns duplex-link-op \$n5 \$n6  
 orient left  
 \$ns duplex-link-op \$n6 \$n7  
 orient left  
 \$ns duplex-link-op \$n7 \$n8  
 orient left  
 \$ns duplex-link-op \$n8 \$n9  
 orient down  
 \$ns duplex-link-op \$n9 \$n10  
 orient down  
 \$ns duplex-link-op \$n10 \$n11  
 orient down  
 \$ns duplex-link-op \$n11 \$n12  
 orient down  
 \$ns duplex-link-op \$n12 \$n13  
 orient down  
 \$ns duplex-link-op \$n13 \$n14  
 orient left  
 \$ns duplex-link-op \$n14 \$n15  
 orient left  
 \$ns duplex-link-op \$n15 \$n16  
 orient left  
 \$ns duplex-link-op \$n16 \$n17  
 orient left  
 \$ns duplex-link-op \$n17 \$n18  
 orient left  
 \$ns duplex-link-op \$n18 \$n19  
 orient left  
 \$ns duplex-link-op \$n19 \$n20  
 orient down  
 \$ns duplex-link-op \$n20 \$n21  
 orient down

```

$ns duplex-link-op $n21 $n22
orient down
$ns duplex-link-op $n22 $n23
orient down
$ns duplex-link-op $n24 $n25
orient up
$ns duplex-link-op $n25 $n26
orient up
$ns duplex-link-op $n26 $n27
orient up
$ns duplex-link-op $n27 $n28
orient up
$ns duplex-link-op $n28 $n29
orient up
$ns duplex-link-op $n29 $n30
orient up-left
$ns duplex-link-op $n30 $n31
orient up
$ns duplex-link-op $n31 $n32
orient up-left
$ns duplex-link-op $n32 $n33
orient left
$ns duplex-link-op $n33 $n34
orient up
$ns duplex-link-op $n34 $n35
orient up
$ns duplex-link-op $n35 $n36
orient up
$ns duplex-link-op $n36 $n37
orient up
$ns duplex-link-op $n37 $n38
orient up
$ns duplex-link-op $n38 $n39
orient up
$ns duplex-link-op $n39 $n40
orient right
$ns duplex-link-op $n40 $n41
orient down

```

```

$ns duplex-link-op $n41 $n42
orient down
$ns duplex-link-op $n42 $n43
orient down
$ns duplex-link-op $n43 $n44
orient down
$ns duplex-link-op $n44 $n45
orient down
$ns duplex-link-op $n45 $n46
orient down-right
$ns duplex-link-op $n46 $n47
orient right
$ns duplex-link-op $n47 $n48
orient down
$ns duplex-link-op $n48 $n49
orient down
$ns duplex-link-op $n49 $n29
orient down
$ns duplex-link-op $n24 $n14
orient down

```

```

#Membuat TCP agent dan
attach di dalam node
set tcp0 [new Agent/TCP]
$ns attach-agent $n15 $tcp0
$tcp0 set packetSize_ 1600
set sink0 [new
Agent/TCPSink]
$ns attach-agent $n14 $sink0
$ns connect $tcp0 $sink0

```

```

#tes menggunakan UDP agent
dengan trafik CBR
#set udp [new Agent/UDP]
#$ns attach-agent $n0 $udp
#set null [new Agent/Null]
#$ns attach-agent $n14 $null
#$ns connect $udp $null
#$udp set fid_ 2

```



```

#agent UDP
#set      cbr      [new
Application/Traffic/CBR]
#$cbr attach-agent $udp
#$cbr set packetSize_ 1600
#$cbr set rate_ 0.01Mb
#$cbr set random_ false

#set tcp1 [new Agent/TCP]
#$ns attach-agent $n39 $tcp1
#set      sink1      [new
Agent/TCPSink]
#$ns attach-agent $n24 $sink1
#$ns connect $tcp1 $sink1

$tcp0 set fid_ 1

set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

#set      cbr0      [new
Application/Traffic/CBR]
#$cbr0 attach-agent $tcp0
#set      cbr1      [new
Application/Traffic/CBR]

```

```

#$cbr1 attach-agent $tcp1

$ns at 0.5 "$ftp0 start"
$ns at 1.0 "$ftp0 stop"

#proc plotWindow {tcpSource
file} {
#global ns
#set time 0.5
#set now [$ns now]
#set cwnd [tcpSource set
cwnd_]
#set wnd [tcpSource set
window_]
#puts $file "$now $cwnd"
#$ns at [expr $now+$time]
"plotWindow      $tcpSource
$file" }
#$ns at 0.5 "plotWindow $cbr
$winfile"

$ns at 5.5 "finish"
puts "starting simulation . . ."
$ns run

```

**Program cek2mono.tcl (pengiriman secara bersamaan dari semua node ke server)**

```
#Create a simulator object
set ns [new Simulator]

#Perbedaan aliran data (NAM)
$ns color 0 blue
$ns color 1 blue
$ns color 2 blue
$ns color 3 blue
$ns color 4 blue
$ns color 5 blue
$ns color 6 blue
$ns color 7 blue
$ns color 8 blue
$ns color 9 blue
$ns color 10 blue
$ns color 11 blue
$ns color 12 blue
$ns color 13 blue
$ns color 14 blue
$ns color 15 blue
$ns color 16 blue
$ns color 17 blue
$ns color 18 blue
$ns color 19 blue
$ns color 20 blue
$ns color 21 blue
$ns color 22 blue
$ns color 23 blue

#Membuka Trace file
set file1 [open outtrace-baru.tr
w]
set winfile [open WinFile w]
$ns trace-all $file1

#Membuka NAM trace file
set file2 [open outtrace-
baru.nam w]
$ns namtrace-all $file2

#Eksekusi terakhir
proc finish {} {
    global ns file1 file2
    $ns flush-trace
    close $file1
    close $file2
    exec nam outtrace-
baru.nam &

    exit 0
}

#Membuat node
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
set n9 [$ns node]
set n10 [$ns node]
set n11 [$ns node]
set n12 [$ns node]
set n13 [$ns node]
set n14 [$ns node]
set n15 [$ns node]
set n16 [$ns node]
set n17 [$ns node]
```

```

set n18 [$ns node]
set n19 [$ns node]
set n20 [$ns node]
set n21 [$ns node]
set n22 [$ns node]
set n23 [$ns node]

```

```

$N14 shape box
$N14 color red

```

```

#membuat label
$N0 label "M-1"
$N1 label "M-2"
$N2 label "M-3"
$N3 label "M-4"
$N4 label "M-5"
$N5 label "M-6"
$N6 label "M-7"
$N7 label "M-8"
$N8 label "M-9"
$N9 label "M-10"
$N10 label "M-11"
$N11 label "M-12"
$N12 label "M-13"
$N13 label "M-14"
$N14 label "M-15"
$N15 label "M-16"
$N16 label "M-17"
$N17 label "M-18"
$N18 label "M-19"
$N19 label "M-20"
$N20 label "M-21"
$N21 label "M-22"
$N22 label "M-23"
$N23 label "M-24"

```

```

#Membuat link diantara node

```

```

$Ns duplex-link $N0 $N1
100Mb 0.0028786129ms
DropTail
$Ns duplex-link $N1 $N2
100Mb 0.0054991937ms
DropTail
$Ns duplex-link $N2 $N3
100Mb 0.0050802225ms
DropTail
$Ns duplex-link $N3 $N4
100Mb 0.0037313155ms
DropTail
$Ns duplex-link $N4 $N5
100Mb 0.0019691746ms
DropTail
$Ns duplex-link $N5 $N6
100Mb 0.0077150730ms
DropTail
$Ns duplex-link $N6 $N7
100Mb 0.0053534070ms
DropTail
$Ns duplex-link $N7 $N8
100Mb 0.0043378038ms
DropTail
$Ns duplex-link $N8 $N9
100Mb 0.0036016620ms
DropTail
$Ns duplex-link $N9 $N10
100Mb 0.0066950565ms
DropTail
$Ns duplex-link $N10 $N11
100Mb 0.0066950074ms
DropTail
$Ns duplex-link $N11 $N12
100Mb 0.0053961181ms
DropTail
$Ns duplex-link $N12 $N13
100Mb 0.0014756183ms
DropTail

```

\$ns duplex-link \$n13 \$n14  
 100Mb 0.0027892677ms  
 DropTail  
 \$ns duplex-link \$n14 \$n15  
 100Mb 0.0038118831ms  
 DropTail  
 \$ns duplex-link \$n15 \$n16  
 100Mb 0.0038829866ms  
 DropTail  
 \$ns duplex-link \$n16 \$n17  
 100Mb 0.0056240416ms  
 DropTail  
 \$ns duplex-link \$n17 \$n18  
 100Mb 0.0047276962ms  
 DropTail  
 \$ns duplex-link \$n18 \$n19  
 100Mb 0.0080848602ms  
 DropTail  
 \$ns duplex-link \$n19 \$n20  
 100Mb 0.0044103784ms  
 DropTail  
 \$ns duplex-link \$n20 \$n21  
 100Mb 0.0046955280ms  
 DropTail  
 \$ns duplex-link \$n21 \$n22  
 100Mb 0.0062420536ms  
 DropTail  
 \$ns duplex-link \$n22 \$n23  
 100Mb 0.0084909838ms  
 DropTail  
  
 \$ns duplex-link-op \$n0 \$n1  
 orient left  
 \$ns duplex-link-op \$n1 \$n2  
 orient left  
 \$ns duplex-link-op \$n2 \$n3  
 orient left  
 \$ns duplex-link-op \$n3 \$n4  
 orient left

\$ns duplex-link-op \$n4 \$n5  
 orient left  
 \$ns duplex-link-op \$n5 \$n6  
 orient left  
 \$ns duplex-link-op \$n6 \$n7  
 orient left  
 \$ns duplex-link-op \$n7 \$n8  
 orient left  
 \$ns duplex-link-op \$n8 \$n9  
 orient down  
 \$ns duplex-link-op \$n9 \$n10  
 orient down  
 \$ns duplex-link-op \$n10 \$n11  
 orient down  
 \$ns duplex-link-op \$n11 \$n12  
 orient down  
 \$ns duplex-link-op \$n12 \$n13  
 orient down  
 \$ns duplex-link-op \$n13 \$n14  
 orient left  
 \$ns duplex-link-op \$n14 \$n15  
 orient left  
 \$ns duplex-link-op \$n15 \$n16  
 orient left  
 \$ns duplex-link-op \$n16 \$n17  
 orient left  
 \$ns duplex-link-op \$n17 \$n18  
 orient left  
 \$ns duplex-link-op \$n18 \$n19  
 orient left  
 \$ns duplex-link-op \$n19 \$n20  
 orient down  
 \$ns duplex-link-op \$n20 \$n21  
 orient down  
 \$ns duplex-link-op \$n21 \$n22  
 orient down  
 \$ns duplex-link-op \$n22 \$n23  
 orient down

```
#Membuat TCP agent dan
attach di dalam node
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
$tcp0 set packetSize_ 100kb
set      sink0      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink0
$ns connect $tcp0 $sink0
```

```
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
$tcp1 set packetSize_ 100kb
set      sink1      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink1
$ns connect $tcp1 $sink1
```

```
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
$tcp2 set packetSize_ 100kb
set      sink2      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink2
$ns connect $tcp2 $sink2
```

```
set tcp3 [new Agent/TCP]
$ns attach-agent $n3 $tcp3
$tcp3 set packetSize_ 100kb
set      sink3      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink3
$ns connect $tcp3 $sink3
```

```
set tcp4 [new Agent/TCP]
$ns attach-agent $n4 $tcp4
$tcp4 set packetSize_ 100kb
set      sink4      [new
Agent/TCPSink]
```

```
$ns attach-agent $n14 $sink4
$ns connect $tcp4 $sink4
```

```
set tcp5 [new Agent/TCP]
$ns attach-agent $n5 $tcp5
$tcp5 set packetSize_ 100kb
set      sink5      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink5
$ns connect $tcp5 $sink5
```

```
set tcp6 [new Agent/TCP]
$ns attach-agent $n6 $tcp6
$tcp6 set packetSize_ 100kb
set      sink6      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink6
$ns connect $tcp6 $sink6
```

```
set tcp7 [new Agent/TCP]
$ns attach-agent $n7 $tcp7
$tcp7 set packetSize_ 100kb
set      sink7      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink7
$ns connect $tcp7 $sink7
```

```
set tcp8 [new Agent/TCP]
$ns attach-agent $n8 $tcp8
$tcp8 set packetSize_ 100kb
set      sink8      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink8
$ns connect $tcp8 $sink8
```

```
set tcp9 [new Agent/TCP]
$ns attach-agent $n9 $tcp9
$tcp9 set packetSize_ 100kb
```

```
set          sink9          [new
Agent/TCPSink]
$ns attach-agent $n14 $sink9
$ns connect $tcp9 $sink9
```

```
set tcp10 [new Agent/TCP]
$ns attach-agent $n10 $tcp10
$tcp10 set packetSize_ 100kb
set          sink10         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink10
$ns connect $tcp10 $sink10
```

```
set tcp11 [new Agent/TCP]
$ns attach-agent $n11 $tcp11
$tcp11 set packetSize_ 100kb
set          sink11         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink11
$ns connect $tcp11 $sink11
```

```
set tcp12 [new Agent/TCP]
$ns attach-agent $n12 $tcp12
$tcp12 set packetSize_ 100kb
set          sink12         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink12
$ns connect $tcp12 $sink12
```

```
set tcp13 [new Agent/TCP]
$ns attach-agent $n13 $tcp13
$tcp13 set packetSize_ 100kb
set          sink13         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink13
$ns connect $tcp13 $sink13
```

```
set tcp15 [new Agent/TCP]
$ns attach-agent $n15 $tcp15
```

```
$tcp15 set packetSize_ 100kb
set          sink15         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink15
$ns connect $tcp15 $sink15
```

```
set tcp16 [new Agent/TCP]
$ns attach-agent $n16 $tcp16
$tcp16 set packetSize_ 100kb
set          sink16         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink16
$ns connect $tcp16 $sink16
```

```
set tcp17 [new Agent/TCP]
$ns attach-agent $n17 $tcp17
$tcp17 set packetSize_ 100kb
set          sink17         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink17
$ns connect $tcp17 $sink17
```

```
set tcp18 [new Agent/TCP]
$ns attach-agent $n18 $tcp18
$tcp18 set packetSize_ 100kb
set          sink18         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink18
$ns connect $tcp18 $sink18
```

```
set tcp19 [new Agent/TCP]
$ns attach-agent $n19 $tcp19
$tcp19 set packetSize_ 100kb
set          sink19         [new
Agent/TCPSink]
$ns attach-agent $n14 $sink19
$ns connect $tcp19 $sink19
```

```
set tcp20 [new Agent/TCP]
```

```

$ns attach-agent $n20 $tcp20
$tcp20 set packetSize_ 100kb
set      sink20      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink20
$ns connect $tcp20 $sink20

```

```

set tcp21 [new Agent/TCP]
$ns attach-agent $n21 $tcp21
$tcp21 set packetSize_ 100kb
set      sink21      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink21
$ns connect $tcp21 $sink21

```

```

set tcp22 [new Agent/TCP]
$ns attach-agent $n22 $tcp22
$tcp22 set packetSize_ 100kb
set      sink22      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink22
$ns connect $tcp22 $sink22

```

```

set tcp23 [new Agent/TCP]
$ns attach-agent $n23 $tcp23
$tcp23 set packetSize_ 100kb
set      sink23      [new
Agent/TCPSink]
$ns attach-agent $n14 $sink23
$ns connect $tcp23 $sink23

```

```

$tcp0 set fid_ 1
$tcp1 set fid_ 2
$tcp2 set fid_ 3
$tcp3 set fid_ 4
$tcp4 set fid_ 5
$tcp5 set fid_ 6
$tcp6 set fid_ 7
$tcp7 set fid_ 8

```

```

$tcp8 set fid_ 9
$tcp9 set fid_ 10
$tcp10 set fid_ 11
$tcp11 set fid_ 12
$tcp12 set fid_ 13
$tcp13 set fid_ 14
$tcp15 set fid_ 16
$tcp16 set fid_ 17
$tcp17 set fid_ 18
$tcp18 set fid_ 19
$tcp19 set fid_ 20
$tcp20 set fid_ 21
$tcp21 set fid_ 22
$tcp22 set fid_ 23
$tcp23 set fid_ 24

```

```

set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

```

```

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1

```

```

set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2

```

```

set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3

```

```

set ftp4 [new Application/FTP]
$ftp4 attach-agent $tcp4

```

```

set ftp5 [new Application/FTP]
$ftp5 attach-agent $tcp5

```

```

set ftp6 [new Application/FTP]
$ftp6 attach-agent $tcp6

```

```

set ftp7 [new Application/FTP]
$ftp7 attach-agent $tcp7

```

set ftp8 [new Application/FTP] \$ftp8 attach-agent \$tcp8	set ftp19 [new Application/FTP] \$ftp19 attach-agent \$tcp19
set ftp9 [new Application/FTP] \$ftp9 attach-agent \$tcp9	set ftp20 [new Application/FTP] \$ftp20 attach-agent \$tcp20
set ftp10 [new Application/FTP] \$ftp10 attach-agent \$tcp10	set ftp21 [new Application/FTP] \$ftp21 attach-agent \$tcp21
set ftp11 [new Application/FTP] \$ftp11 attach-agent \$tcp11	set ftp22 [new Application/FTP] \$ftp22 attach-agent \$tcp22
set ftp12 [new Application/FTP] \$ftp12 attach-agent \$tcp12	set ftp23 [new Application/FTP] \$ftp23 attach-agent \$tcp23
set ftp13 [new Application/FTP] \$ftp13 attach-agent \$tcp13	\$ns at 0.5 "\$ftp0 start"
set ftp15 [new Application/FTP] \$ftp15 attach-agent \$tcp15	\$ns at 0.5 "\$ftp1 start"
set ftp16 [new Application/FTP] \$ftp16 attach-agent \$tcp16	\$ns at 0.5 "\$ftp2 start"
set ftp17 [new Application/FTP] \$ftp17 attach-agent \$tcp17	\$ns at 0.5 "\$ftp3 start"
set ftp18 [new Application/FTP] \$ftp18 attach-agent \$tcp18	\$ns at 0.5 "\$ftp4 start"
	\$ns at 0.5 "\$ftp5 start"
	\$ns at 0.5 "\$ftp6 start"
	\$ns at 0.5 "\$ftp7 start"
	\$ns at 0.5 "\$ftp8 start"
	\$ns at 0.5 "\$ftp9 start"
	\$ns at 0.5 "\$ftp10 start"
	\$ns at 0.5 "\$ftp11 start"
	\$ns at 0.5 "\$ftp12 start"
	\$ns at 0.5 "\$ftp13 start"
	\$ns at 0.5 "\$ftp15 start"
	\$ns at 0.5 "\$ftp16 start"
	\$ns at 0.5 "\$ftp17 start"
	\$ns at 0.5 "\$ftp18 start"
	\$ns at 0.5 "\$ftp19 start"



```
$ns at 0.5 "$ftp20 start"
$ns at 0.5 "$ftp21 start"
$ns at 0.5 "$ftp22 start"
$ns at 0.5 "$ftp23 start"
```

```
$ns at 1.5 "$ftp20 stop"
$ns at 1.5 "$ftp21 stop"
$ns at 1.5 "$ftp22 stop"
$ns at 1.5 "$ftp23 stop"
```

```
$ns at 1.5 "$ftp0 stop"
$ns at 1.5 "$ftp1 stop"
$ns at 1.5 "$ftp2 stop"
$ns at 1.5 "$ftp3 stop"
$ns at 1.5 "$ftp4 stop"
$ns at 1.5 "$ftp5 stop"
$ns at 1.5 "$ftp6 stop"
$ns at 1.5 "$ftp7 stop"
$ns at 1.5 "$ftp8 stop"
$ns at 1.5 "$ftp9 stop"
$ns at 1.5 "$ftp10 stop"
$ns at 1.5 "$ftp11 stop"
$ns at 1.5 "$ftp12 stop"
$ns at 1.5 "$ftp13 stop"
$ns at 1.5 "$ftp15 stop"
$ns at 1.5 "$ftp16 stop"
$ns at 1.5 "$ftp17 stop"
$ns at 1.5 "$ftp18 stop"
$ns at 1.5 "$ftp19 stop"
```

```
#proc plotWindow {tcpSource
file} {
#global ns
#set time 0.5
#set now [$ns now]
#set cwnd [StcpSource set
cwnd_]
#set wnd [StcpSource set
window_]
#puts $file "$now $cwnd"
#$ns at [expr $now+$time]
"plotWindow $tcpSource
$file" }
#$ns at 0.5 "plotWindow $tcp0
$winfile"
```

```
$ns at 5.5 "finish"
puts "starting simulation . . ."
$ns run
```

### **Program cek2tram.tcl (pengiriman secara bersamaan dari semua node ke server)**

```
#Create a simulator object
set ns [new Simulator]
```

```
#Perbedaan aliran data (NAM)
```

```
$ns color 0 blue
$ns color 1 blue
$ns color 2 blue
$ns color 3 blue
$ns color 4 blue
```

```
$ns color 5 blue
$ns color 6 blue
$ns color 7 blue
$ns color 8 blue
$ns color 9 blue
$ns color 10 blue
$ns color 11 blue
$ns color 12 blue
$ns color 13 blue
```

```

$ns color 14 blue
$ns color 15 blue
$ns color 16 blue
$ns color 17 blue
$ns color 18 blue
$ns color 19 blue
$ns color 20 blue
$ns color 21 blue
$ns color 22 blue
$ns color 23 blue
$ns color 24 blue
$ns color 25 blue
$ns color 26 blue
$ns color 27 blue
$ns color 28 blue
$ns color 29 blue
$ns color 30 blue
$ns color 31 blue
$ns color 32 blue
$ns color 33 blue
$ns color 34 blue
$ns color 35 blue
$ns color 36 blue
$ns color 37 blue
$ns color 38 blue
$ns color 39 blue
$ns color 40 blue
$ns color 41 blue
$ns color 42 blue
$ns color 43 blue
$ns color 44 blue
$ns color 45 blue
$ns color 46 blue
$ns color 47 blue
$ns color 48 blue
$ns color 49 blue

#Membuka Trace file

```

```

set file1 [open outtrace-baru.tr
w]
set winfile [open WinFile w]
$ns trace-all $file1

#Membuka NAM trace file
set file2 [open outtrace-
baru.nam w]
$ns namtrace-all $file2

#Eksekusi terakhir
proc finish {} {
    global ns file1 file2
    $ns flush-trace
    close $file1
    close $file2
    exec nam outtrace-
baru.nam &

    exit 0
}

set n24 [$ns node]
set n25 [$ns node]
set n26 [$ns node]
set n27 [$ns node]
set n28 [$ns node]
set n29 [$ns node]
set n30 [$ns node]
set n31 [$ns node]
set n32 [$ns node]
set n33 [$ns node]
set n34 [$ns node]
set n35 [$ns node]
set n36 [$ns node]
set n37 [$ns node]
set n38 [$ns node]
set n39 [$ns node]
set n40 [$ns node]

```

```

set n41 [$ns node]
set n42 [$ns node]
set n43 [$ns node]
set n44 [$ns node]
set n45 [$ns node]
set n46 [$ns node]
set n47 [$ns node]
set n48 [$ns node]
set n49 [$ns node]

```

```

$N24 shape box
$N24 color red

```

```

$N24 label "T-1"
$N25 label "T-2"
$N26 label "T-3"
$N27 label "T-4"
$N28 label "T-5"
$N29 label "T-6"
$N30 label "T-7"
$N31 label "T-8"
$N32 label "T-9"
$N33 label "T-10"
$N34 label "T-11"
$N35 label "T-12"
$N36 label "T-13"
$N37 label "T-14"
$N38 label "T-15"
$N39 label "T-16"
$N40 label "T-17"
$N41 label "T-18"
$N42 label "T-19"
$N43 label "T-20"
$N44 label "T-21"
$N45 label "T-22"
$N46 label "T-23"
$N47 label "T-24"
$N48 label "T-25"

```

```

$N49 label "T-26"

```

```

$ns duplex-link $N24 $N25
100Mb      0.0014004938ms
DropTail
$ns duplex-link $N25 $N26
100Mb      0.0022935537ms
DropTail
$ns duplex-link $N26 $N27
100Mb      0.0025271655ms
DropTail
$ns duplex-link $N27 $N28
100Mb      0.0038853404ms
DropTail
$ns duplex-link $N28 $N29
100Mb      0.0039194210ms
DropTail
$ns duplex-link $N29 $N30
100Mb      0.0023367552ms
DropTail
$ns duplex-link $N30 $N31
100Mb      0.0021312906ms
DropTail
$ns duplex-link $N31 $N32
100Mb      0.0035770454ms
DropTail
$ns duplex-link $N32 $N33
100Mb      0.0023596554ms
DropTail
$ns duplex-link $N33 $N34
100Mb      0.0025697295ms
DropTail
$ns duplex-link $N34 $N35
100Mb      0.0018477593ms
DropTail
$ns duplex-link $N35 $N36
100Mb      0.0030509286ms
DropTail

```

\$ns duplex-link \$n36 \$n37  
 100Mb 0.0021139316ms  
 DropTail  
 \$ns duplex-link \$n37 \$n38  
 100Mb 0.0027449874ms  
 DropTail  
 \$ns duplex-link \$n38 \$n39  
 100Mb 0.0019162148ms  
 DropTail  
 \$ns duplex-link \$n39 \$n40  
 100Mb 0.0050670806ms  
 DropTail  
 \$ns duplex-link \$n40 \$n41  
 100Mb 0.0028598318ms  
 DropTail  
 \$ns duplex-link \$n41 \$n42  
 100Mb 0.0024900936ms  
 DropTail  
 \$ns duplex-link \$n42 \$n43  
 100Mb 0.0027668088ms  
 DropTail  
 \$ns duplex-link \$n43 \$n44  
 100Mb 0.0020487615ms  
 DropTail  
 \$ns duplex-link \$n44 \$n45  
 100Mb 0.0036068108ms  
 DropTail  
 \$ns duplex-link \$n45 \$n46  
 100Mb 0.0014419790ms  
 DropTail  
 \$ns duplex-link \$n46 \$n47  
 100Mb 0.0031695488ms  
 DropTail  
 \$ns duplex-link \$n47 \$n48  
 100Mb 0.0020614621ms  
 DropTail  
 \$ns duplex-link \$n48 \$n49  
 100Mb 0.0020154164ms  
 DropTail

\$ns duplex-link \$n49 \$n29  
 100Mb 0.0018470237ms  
 DropTail  
  
 \$ns duplex-link-op \$n24 \$n25  
 orient up  
 \$ns duplex-link-op \$n25 \$n26  
 orient up  
 \$ns duplex-link-op \$n26 \$n27  
 orient up  
 \$ns duplex-link-op \$n27 \$n28  
 orient up  
 \$ns duplex-link-op \$n28 \$n29  
 orient up  
 \$ns duplex-link-op \$n29 \$n30  
 orient up-left  
 \$ns duplex-link-op \$n30 \$n31  
 orient up  
 \$ns duplex-link-op \$n31 \$n32  
 orient up-left  
 \$ns duplex-link-op \$n32 \$n33  
 orient left  
 \$ns duplex-link-op \$n33 \$n34  
 orient up  
 \$ns duplex-link-op \$n34 \$n35  
 orient up  
 \$ns duplex-link-op \$n35 \$n36  
 orient up  
 \$ns duplex-link-op \$n36 \$n37  
 orient up  
 \$ns duplex-link-op \$n37 \$n38  
 orient up  
 \$ns duplex-link-op \$n38 \$n39  
 orient up  
 \$ns duplex-link-op \$n39 \$n40  
 orient right  
 \$ns duplex-link-op \$n40 \$n41  
 orient down

```

$ns duplex-link-op $n41 $n42
orient down
$ns duplex-link-op $n42 $n43
orient down
$ns duplex-link-op $n43 $n44
orient down
$ns duplex-link-op $n44 $n45
orient down
$ns duplex-link-op $n45 $n46
orient down-right
$ns duplex-link-op $n46 $n47
orient right
$ns duplex-link-op $n47 $n48
orient down
$ns duplex-link-op $n48 $n49
orient down
$ns duplex-link-op $n49 $n29
orient down

```

```

#jalur tram
set tcp25 [new Agent/TCP]
$ns attach-agent $n25 $tcp25
$tcp25 set packetSize_ 100kb
set sink25 [new Agent/TCPSink]
$ns attach-agent $n24 $sink25
$ns connect $tcp25 $sink25

```

```

set tcp26 [new Agent/TCP]
$ns attach-agent $n26 $tcp26
$tcp26 set packetSize_ 100kb
set sink26 [new Agent/TCPSink]
$ns attach-agent $n24 $sink26
$ns connect $tcp26 $sink26

```

```

set tcp27 [new Agent/TCP]
$ns attach-agent $n27 $tcp27

```

```

$tcp27 set packetSize_ 100kb
set sink27 [new Agent/TCPSink]
$ns attach-agent $n24 $sink27
$ns connect $tcp27 $sink27

```

```

set tcp28 [new Agent/TCP]
$ns attach-agent $n28 $tcp28
$tcp28 set packetSize_ 100kb
set sink28 [new Agent/TCPSink]
$ns attach-agent $n24 $sink28
$ns connect $tcp28 $sink28

```

```

set tcp29 [new Agent/TCP]
$ns attach-agent $n29 $tcp29
$tcp29 set packetSize_ 100kb
set sink29 [new Agent/TCPSink]
$ns attach-agent $n24 $sink29
$ns connect $tcp29 $sink29

```

```

set tcp30 [new Agent/TCP]
$ns attach-agent $n30 $tcp30
$tcp30 set packetSize_ 100kb
set sink30 [new Agent/TCPSink]
$ns attach-agent $n24 $sink30
$ns connect $tcp30 $sink30

```

```

set tcp31 [new Agent/TCP]
$ns attach-agent $n31 $tcp31
$tcp31 set packetSize_ 100kb
set sink31 [new Agent/TCPSink]
$ns attach-agent $n24 $sink31
$ns connect $tcp31 $sink31

```

```

set tcp32 [new Agent/TCP]

```

```
$ns attach-agent $n32 $tcp32
$tcp32 set packetSize_ 100kb
set      sink32      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink32
$ns connect $tcp32 $sink32
```

```
set tcp33 [new Agent/TCP]
$ns attach-agent $n33 $tcp33
$tcp33 set packetSize_ 100kb
set      sink33      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink33
$ns connect $tcp33 $sink33
```

```
set tcp34 [new Agent/TCP]
$ns attach-agent $n34 $tcp34
$tcp34 set packetSize_ 100kb
set      sink34      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink34
$ns connect $tcp34 $sink34
```

```
set tcp35 [new Agent/TCP]
$ns attach-agent $n35 $tcp35
$tcp35 set packetSize_ 100kb
set      sink35      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink35
$ns connect $tcp35 $sink35
```

```
set tcp36 [new Agent/TCP]
$ns attach-agent $n36 $tcp36
$tcp36 set packetSize_ 100kb
set      sink36      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink36
$ns connect $tcp36 $sink36
```

```
set tcp37 [new Agent/TCP]
$ns attach-agent $n37 $tcp37
$tcp37 set packetSize_ 100kb
set      sink37      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink37
$ns connect $tcp37 $sink37
```

```
set tcp38 [new Agent/TCP]
$ns attach-agent $n38 $tcp38
$tcp38 set packetSize_ 100kb
set      sink38      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink38
$ns connect $tcp38 $sink38
```

```
set tcp39 [new Agent/TCP]
$ns attach-agent $n39 $tcp39
$tcp39 set packetSize_ 100kb
set      sink39      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink39
$ns connect $tcp39 $sink39
```

```
set tcp40 [new Agent/TCP]
$ns attach-agent $n40 $tcp40
$tcp40 set packetSize_ 100kb
set      sink40      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink40
$ns connect $tcp40 $sink40
```

```
set tcp41 [new Agent/TCP]
$ns attach-agent $n41 $tcp41
$tcp41 set packetSize_ 100kb
set      sink41      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink41
$ns connect $tcp41 $sink41
```

```

set tcp42 [new Agent/TCP]
$ns attach-agent $n42 $tcp42
$tcp42 set packetSize_ 100kb
set      sink42      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink42
$ns connect $tcp42 $sink42

```

```

set tcp43 [new Agent/TCP]
$ns attach-agent $n43 $tcp43
$tcp43 set packetSize_ 100kb
set      sink43      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink43
$ns connect $tcp43 $sink43

```

```

set tcp44 [new Agent/TCP]
$ns attach-agent $n44 $tcp44
$tcp44 set packetSize_ 100kb
set      sink44      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink44
$ns connect $tcp44 $sink44

```

```

set tcp45 [new Agent/TCP]
$ns attach-agent $n45 $tcp45
$tcp45 set packetSize_ 100kb
set      sink45      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink45
$ns connect $tcp45 $sink45

```

```

set tcp46 [new Agent/TCP]
$ns attach-agent $n46 $tcp46
$tcp46 set packetSize_ 100kb
set      sink46      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink46

```

```

$ns connect $tcp46 $sink46

```

```

set tcp47 [new Agent/TCP]
$ns attach-agent $n47 $tcp47
$tcp47 set packetSize_ 100kb
set      sink47      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink47
$ns connect $tcp47 $sink47

```

```

set tcp48 [new Agent/TCP]
$ns attach-agent $n48 $tcp48
$tcp48 set packetSize_ 100kb
set      sink48      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink48
$ns connect $tcp48 $sink48

```

```

set tcp49 [new Agent/TCP]
$ns attach-agent $n49 $tcp49
$tcp49 set packetSize_ 100kb
set      sink49      [new
Agent/TCPSink]
$ns attach-agent $n24 $sink49
$ns connect $tcp49 $sink49

```

```

#tes menggunakan UDP agent
dengan trafik CBR

```

```

#set udp [new Agent/UDP]
#$ns attach-agent $n0 $udp
#set null [new Agent/Null]
#$ns attach-agent $n14 $null
#$ns connect $udp $null
#$udp set fid_ 2

```

```

#agent UDP
#set      cbr      [new
Application/Traffic/CBR]
#$cbr attach-agent $udp

```

```

#$cbr set packetSize_ 1600
#$cbr set rate_ 0.01Mb
#$cbr set random_ false

#set tcp1 [new Agent/TCP]
#sns attach-agent $n39 $tcp1
#set      sink1      [new
Agent/TCPSink]
#sns attach-agent $n24 $sink1
#sns connect $tcp1 $sink1

```

```

$tcp25 set fid_ 26
$tcp26 set fid_ 27
$tcp27 set fid_ 28
$tcp28 set fid_ 29
$tcp29 set fid_ 30
$tcp30 set fid_ 31
$tcp31 set fid_ 32
$tcp32 set fid_ 33
$tcp33 set fid_ 34
$tcp34 set fid_ 35
$tcp35 set fid_ 36
$tcp36 set fid_ 37
$tcp37 set fid_ 38
$tcp38 set fid_ 39
$tcp39 set fid_ 40
$tcp40 set fid_ 41
$tcp41 set fid_ 42
$tcp42 set fid_ 43
$tcp43 set fid_ 44
$tcp44 set fid_ 45
$tcp45 set fid_ 46
$tcp46 set fid_ 47
$tcp47 set fid_ 48
$tcp48 set fid_ 49
$tcp49 set fid_ 50

```

```

set      ftp25      [new
Application/FTP]

```

```

$ftp25 attach-agent $tcp25

set      ftp26      [new
Application/FTP]
$ftp26 attach-agent $tcp26

set      ftp27      [new
Application/FTP]
$ftp27 attach-agent $tcp27

set      ftp28      [new
Application/FTP]
$ftp28 attach-agent $tcp28

set      ftp29      [new
Application/FTP]
$ftp29 attach-agent $tcp29

set      ftp30      [new
Application/FTP]
$ftp30 attach-agent $tcp30

set      ftp31      [new
Application/FTP]
$ftp31 attach-agent $tcp31

set      ftp32      [new
Application/FTP]
$ftp32 attach-agent $tcp32

set      ftp33      [new
Application/FTP]
$ftp33 attach-agent $tcp33

set      ftp34      [new
Application/FTP]
$ftp34 attach-agent $tcp34

```



```

set      ftp35      [new
Application/FTP]
$ftp35 attach-agent $tcp35

set      ftp36      [new
Application/FTP]
$ftp36 attach-agent $tcp36

set      ftp37      [new
Application/FTP]
$ftp37 attach-agent $tcp37

set      ftp38      [new
Application/FTP]
$ftp38 attach-agent $tcp38

set      ftp39      [new
Application/FTP]
$ftp39 attach-agent $tcp39

set      ftp40      [new
Application/FTP]
$ftp40 attach-agent $tcp40

set      ftp41      [new
Application/FTP]
$ftp41 attach-agent $tcp41

set      ftp42      [new
Application/FTP]
$ftp42 attach-agent $tcp42

set      ftp43      [new
Application/FTP]
$ftp43 attach-agent $tcp43

set      ftp44      [new
Application/FTP]
$ftp44 attach-agent $tcp44

```

```

set      ftp45      [new
Application/FTP]
$ftp45 attach-agent $tcp45

set      ftp46      [new
Application/FTP]
$ftp46 attach-agent $tcp46

set      ftp47      [new
Application/FTP]
$ftp47 attach-agent $tcp47

set      ftp48      [new
Application/FTP]
$ftp48 attach-agent $tcp48

set      ftp49      [new
Application/FTP]
$ftp49 attach-agent $tcp49

#set     cbr0       [new
Application/Traffic/CBR]
#$cbr0 attach-agent $tcp0
#set     cbr1       [new
Application/Traffic/CBR]
#$cbr1 attach-agent $tcp1

$ns at 0.5 "$ftp25 start"
$ns at 0.5 "$ftp26 start"
$ns at 0.5 "$ftp27 start"
$ns at 0.5 "$ftp28 start"
$ns at 0.5 "$ftp29 start"
$ns at 0.5 "$ftp30 start"
$ns at 0.5 "$ftp31 start"
$ns at 0.5 "$ftp32 start"
$ns at 0.5 "$ftp33 start"
$ns at 0.5 "$ftp34 start"

```

```

$ns at 0.5 "$ftp35 start"
$ns at 0.5 "$ftp36 start"
$ns at 0.5 "$ftp37 start"
$ns at 0.5 "$ftp38 start"
$ns at 0.5 "$ftp39 start"
$ns at 0.5 "$ftp40 start"
$ns at 0.5 "$ftp41 start"
$ns at 0.5 "$ftp42 start"
$ns at 0.5 "$ftp43 start"
$ns at 0.5 "$ftp44 start"
$ns at 0.5 "$ftp45 start"
$ns at 0.5 "$ftp46 start"
$ns at 0.5 "$ftp47 start"
$ns at 0.5 "$ftp48 start"
$ns at 0.5 "$ftp49 start"

```

```

$ns at 1.5 "$ftp25 stop"
$ns at 1.5 "$ftp26 stop"
$ns at 1.5 "$ftp27 stop"
$ns at 1.5 "$ftp28 stop"
$ns at 1.5 "$ftp29 stop"
$ns at 1.5 "$ftp30 stop"
$ns at 1.5 "$ftp31 stop"
$ns at 1.5 "$ftp32 stop"
$ns at 1.5 "$ftp33 stop"
$ns at 1.5 "$ftp34 stop"
$ns at 1.5 "$ftp35 stop"
$ns at 1.5 "$ftp36 stop"
$ns at 1.5 "$ftp37 stop"
$ns at 1.5 "$ftp38 stop"
$ns at 1.5 "$ftp39 stop"

```

```

$ns at 1.5 "$ftp40 stop"
$ns at 1.5 "$ftp41 stop"
$ns at 1.5 "$ftp42 stop"
$ns at 1.5 "$ftp43 stop"
$ns at 1.5 "$ftp44 stop"
$ns at 1.5 "$ftp45 stop"
$ns at 1.5 "$ftp46 stop"
$ns at 1.5 "$ftp47 stop"
$ns at 1.5 "$ftp48 stop"
$ns at 1.5 "$ftp49 stop"

```

```

#proc plotWindow {tcpSource
file} {
#global ns
#set time 0.5
#set now [$ns now]
#set cwnd [$tcpSource set
cwnd_]
#set wnd [$tcpSource set
window_]
#puts $file "$now $cwnd"
#$ns at [expr $now+$time]
"plotWindow $tcpSource
$file" }
#$ns at 0.5 "plotWindow $cbr
$winfile"

```

```

$ns at 5.5 "finish"
puts "starting simulation . . ."
$ns run

```

### **-throughput.awk**

```
BEGIN {
  fromNode=15; toNode=14;
  lineCount = 0;totalBits = 0;
}
/^r/&&$3==fromNode&&$4==toNode {
  totalBits += 8*$6;
  if ( lineCount==0 ) {
    timeBegin = $2; lineCount++;
  } else {
    timeEnd = $2;
  };
};
END{
  duration = timeEnd-timeBegin;
  print "Number of records is " NR;
  print "Output: ";
  print "Transmission: N" fromNode "->N" toNode;
  print " - Total transmitted bits = " totalBits " bits";
  print " - duration = " duration " s";
  print " - Throughput = " totalBits/duration/1e3 " kbps.";
};
```

### **-jitter.awk**

```
#This program is used to calculate the jitters for CBR
#  $jitter = ((recvtime(j)-sendtime(j))-(recvtime(i)-sendtime(i)))/(j-i), j > i$ 
BEGIN {
  # Initialization
  highest_packet_id = 0;
}
{
  action = $1;
  time = $2;
  from = $3;
  to = $4;
  type = $5;
  pktsize = $6;
  flow_id = $8;
```

```

src = $9;
dst = $10;
seq_no = $11;
packet_id = $12;
if ( packet_id > highest_packet_id ) {
highest_packet_id = packet_id;
}
#Record the transmission time
if ( start_time[packet_id] == 0 ) {
# Record the sequence number
pkt_seqno[packet_id] = seq_no;
start_time[packet_id] = time;
}
#Record the receiving time for CBR (flow_id=2)
if ( flow_id == 2 && action != "d" ) {
if ( action == "r" ) {
end_time[packet_id] = time;
}
} else {
end_time[packet_id] = -1;
}
}
END {
last_seqno = 0;
last_delay = 0;
seqno_diff = 0;
for ( packet_id = 0; packet_id <= highest_packet_id; packet_id++ ) {
start = start_time[packet_id];
end = end_time[packet_id];
packet_duration = end-start;
if ( start < end ) {
seqno_diff = pkt_seqno[packet_id]-last_seqno;
delay_diff = packet_duration-last_delay;
if (seqno_diff == 0) {
jitter = 0;
} else {
jitter = delay_diff/seqno_diff;
}
}
printf("%f %f\n", start, jitter);

```

```

last_seqno = pkt_seqno[packet_id];
last_delay = packet_duration;
}
}
}

```

### **-Delay.awk**

```

BEGIN {
  for (i in send) {
    send[i] = 0
    sendt[i] = 0
  }
  for (i in recv) {
    recv[i] = 0
    recvt[i] = 0
  }
  delay = 0
  num = 0
  avg_delay = 0
}
{
  # Trace line format: normal
  if ($2 != "-t") {
    event = $1
    time = $2
    node_id_s = $3
    node_id_d = $4
    pkt_type = $5
    pkt_size = $6
    pkt_attrib = $7
    pkt_id = $12
  }
  # Trace line format: new
  #if ($2 == "-t") {
  #   event = $1
  #   time = $3
  #   node_id = $5
  #   flow_id = $39
  #   pkt_id = $41

```

```

#}
# Store packets sent
if (event == "+" && node_id_s == "21" && pkt_type == "tcp") {
send[pkt_id] = time
sendt[pkt_id] = 1
#   print("send[“,pkt_id,”] = “,time)
}
# Store packets arrival time
if (event == "r" && node_id_d == "0" && pkt_type == "tcp") {
recv[pkt_id] = time
recvt[pkt_id] = 1
#   print("recv[“,pkt_id,”] = “,time)
#   print(" -> delay[“,pkt_id,”] = “,recv[pkt_id]-send[pkt_id])
if (recvt[pkt_id] == 1 && sendt[pkt_id] == 1) {
print (time, " ",(recv[pkt_id]-send[pkt_id])) > "delay.tr" }
}
}
END {
# Compute average delay
for (i in recv) {
if (sendt[i] == 1 && recvt[i] == 1) {
delay += recv[i] - send[i]
num ++
}
}
if (num != 0) {
avg_delay = delay / num
} else {
avg_delay = 0
}
print("==> Average delay = ",avg_delay,"s")
print("          = ",avg_delay*1000,"ms")
}

```

#### D. Listing program *server* Xampp

##### -db.php

```
<?php
mysql_connect("localhost","root","");
mysql_select_db("simulasi");
?>
```

##### -tambahsaldo.php

```
<?php
include "db.php";
error_reporting(0);

$id=$_GET['id'];
$jum=$_GET['jum'];

$query = "SELECT saldo FROM tiket WHERE id='$id'";
$result = mysql_query($query) or die ("Error in query: $query.
".mysql_error());
$data=mysql_fetch_array($result);

if ($jum==5) {
    # code...
    $saldoakhir=$data[saldo]+5000;
} else {
    # code...
    $saldoakhir=$data[saldo]+10000;
}

echo "$saldoakhir";

$query="UPDATE tiket SET `saldo` = $saldoakhir where `id`=$id";
$result = mysql_query($query) or die ("Error in query: $query.
".mysql_error());

echo "<script>document.location.href='tiket.php';</script>\n";
?>
```

### **-tap\_in.php**

```
<?php
include "db.php";
error_reporting(0);

$id=$_POST['id'];
$halte=$_POST['halte'];

#ambil data ID kartu lalu mengurangkan nya dengan tiket
$query = "SELECT * FROM tiket WHERE id='$id' ";
$result = mysql_query($query) or die ("Error in query: $query.
".mysql_error());
$data=mysql_fetch_array($result);
if ($data[saldo]<5000) {
    print "<script>alert('Maaf, Saldo anda tidak cukup untuk
melakukan transaksi ini');
    javascript:history.go(-1);</script>";
} else {
    if ($data[id] <= 1000 ) {
        echo "id = $data[id]";
        echo "<br>";
        echo "saldo awal = $data[saldo]";
        $hasil=5000*(50/100);
        echo "<br>";
        echo "harga tiket = $hasil";
        $hasil1=$data[saldo]-$hasil;
        echo "<br>";
        echo "saldo akhir = $hasil1";
    } elseif ($data[id] <= 2000 && $data[id] >= 1001) {
        echo "id = $data[id]";
        echo "<br>";
        echo "saldo awal = $data[saldo]";
        $hasil=5000*(100/100);
        echo "<br>";
        echo "harga tiket = $hasil";
        $hasil1=$data[saldo]-$hasil;
        echo "<br>";
        echo "saldo akhir = $hasil1";
    } elseif ($data[id] <= 3000 && $data[id] >= 2001) {
```



```

        echo "id = $data[id]";
        echo "<br>";
        echo "saldo awal = $data[saldo]";
        $hasil=5000*(75/100);
        echo "<br>";
        echo "harga tiket = $hasil";
        $hasil1=$data[saldo]-$hasil;
        echo "<br>";
        echo "saldo akhir = $hasil1";
    }
    $h1 = "SELECT * FROM halte Where id=$halte";
    $h2 = mysql_query($h1) or die ("Error in query: $h1.
".mysql_error());
    $h3 = mysql_fetch_array($h2);
    $k1 = "SELECT * FROM `kendaraan` Where
`kend`='$h3[jalur]'";
    $k2 = mysql_query($k1) or die ("Error in query: $k1.
".mysql_error());
    $k3 = mysql_fetch_array($k2);
    $totuser=$k3[total]+1;
    $insert1 = "UPDATE kendaraan SET `total`= $totuser,
`latitude`= $h3[lat], `longitude`= $h3[lon], `waktu`= now() WHERE
`id`= $k3[id]";
    $resultinsert1 = mysql_query($insert1) or die ("Error in
query: $insert1. ".mysql_error());
    $insert2 = "UPDATE tiket SET `saldo`=$hasil1 WHERE
`id`=$data[id]";
    $resultinsert2 = mysql_query($insert2) or die ("Error in
query: $insert2. ".mysql_error());
    $rekap = "INSERT INTO `rekap` VALUES (NULL,
'$k3[kend]', '$id', '$h3[lat]', '$h3[lon]', now(), '1', '0')";
    $rekap1 = mysql_query($rekap) or die ("Error in query:
$rekap. ".mysql_error());
    $rekapakhir = "INSERT INTO `rekap2` VALUES (NULL,
'$k3[kend]', '$id', '$h3[lat]', '$h3[lon]', now(), '1', '0')";
    $rekapakhir1 = mysql_query($rekapakhir) or die ("Error in
query: $rekapakhir. ".mysql_error());
}
//echo "$usertot";

```

```

//$querya = "UPDATE kendaraan set `total`= $usertot , `latitude`=
$latitude , `longitude`= $longitude , `waktu`= now() WHERE
`id`=$ambil2[id]";
//$resulta = mysql_query($querya) or die ("Error in query: $querya.
".mysql_error());

//$rekap = "INSERT INTO `rekap` VALUES (NULL, '$plat', '$id',
'$latitude', '$longitude', now(), '1', '0')";
//$rekap1 = mysql_query($rekap) or die ("Error in query: $rekap.
".mysql_error());

//echo
target=_blank>document.location.href='.php';</script>\n";
?>

```

### **-tap\_out.php**

```

<?php
include "db.php";
error_reporting(0);
//halte kartu

$kartu=$_POST['kartu'];
$halte=$_POST['halte'];

//echo "halte = $halte";
echo "<br>";

$ambil= "SELECT * FROM rekap WHERE kartu='$kartu' " ;
$ambil1= mysql_query($ambil) or die($ambil);
$ambil2= mysql_fetch_array($ambil1);
//echo "$ambil2[id]";

$kend= "SELECT * FROM kendaraan WHERE
kend='$ambil2[kend]'";
$kend1= mysql_query($kend) or die($kend);
$array= mysql_fetch_array($kend1);

$halte = "SELECT * FROM halte WHERE id=$halte";
$halte1 = mysql_query($halte);

```

```

$arrayh = mysql_fetch_array($halte1);

$usertot= $array[total]-1;

$querya = "UPDATE kendaraan set `total`= $usertot , `latitude`=
'$arrayh[lat]' , `longitude`= '$arrayh[lon]' , `waktu`= now() WHERE
`id`='$array[id]";
$resulta = mysql_query($querya) or die ("Error in query: $querya.
".mysql_error());

$delete = "DELETE from rekap where id='$sambil2[id]'";
$querydelete= mysql_query($delete);

$rekap = "INSERT INTO `rekap2` VALUES (NULL,
'$sambil2[kend]', '$kartu', '$arrayh[lat]', '$arrayh[lon]', now(), '0', '1')";
$rekap1 = mysql_query($rekap) or die ("Error in query: $rekap.
".mysql_error());

echo "Selamat, anda berhasil melakukan tap-out";
?>

```

### **-maps.php**

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-
scalable=no">
    <meta charset="utf-8">
    <title>Simple markers</title>
    <style>
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
      #map {
        height: 100%;
      }
    </style>

```

```

</style>
</head>
<body>
  <div id="map"></div>
  <?php
    $lat= $_GET['lat'];
    $lng= $_GET['lng'];
    $kend= $_GET['k'];
    if ($kend=="tram") {
      # code...
      $latcen= -7.267055;
      $lngcen= 112.746935;
    } else {
      # code...
      $latcen= -7.307070;
      $lngcen= 112.749338;
    }

    ?>
    <script type="text/javascript">

function initMap() {
  var myLatLng = new google.maps.LatLng(<?php echo $lat;
?>,<?php echo $lng; ?>);

  var myCenter = new google.maps.LatLng(<?php echo $latcen;
?>,<?php echo $lngcen; ?>);

  var myOptions = {
    zoom: 13,
    center: myCenter,
  }
  var map = new
google.maps.Map(document.getElementById("map"), myOptions);

  var marker = new google.maps.Marker({
    position: myLatLng,
    map: map,

```

```

        title:"kendaraan"
    });
}

</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD1_mMm
a0KfxEgQ8rAjGrS2lgcqIqNlGJ8&signed_in=true&callback=initMa
p"></script>
</body>
</html>

```

### **-Halaman ‘Update Live’**

```

<!DOCTYPE html>
<html>
<head>
    <title>Simulasi e-ticketing</title>
</head>
<body>
<table>
<tr>
<td>
<form action="index.php">
    <input type="submit" value="Halaman Utama">
</form></td>
<td>
<form action="tiket.php">
    <input type="submit" value="Daftar Tiket">
</form></td>
<td>
<form action="live.php">
    <input type="submit" value="Update Live">
</form></td>
<td>
<form action="rekap2.php">
    <input type="submit" value="Rekap Akhir">
</form></td></tr>
</table>

```

```

<h2>Daftar kendaraan yang sedang beroperasi</h2>
<?php
include "db.php";
error_reporting(0);
// jumlah data yang akan ditampilkan per halaman

$dataPerPage = 10;

// apabila $_GET['page'] sudah didefinisikan, gunakan nomor halaman
tersebut,
// sedangkan apabila belum, nomor halamannya 1.

if(isset($_GET['paging']))
{
    $noPage = $_GET['paging'];
}
else $noPage = 1;

// perhitungan offset

$offset = ($noPage - 1) * $dataPerPage;

$query=mysql_query("select*from kendaraan LIMIT $offset,
$dataPerPage");
echo "





```

```

echo "<td>".$data['latitude']. "</td>";
echo "<td>".$data['longitude']. "</td>";
echo "<td align=center>".$data['total']. "</td>";
echo "<td width=175 align=center>".$data['waktu']. "</td>";
echo
            "<td
                                align=center><a
href=maps.php?lat=".$data['latitude']."&lng=".$data['longitude']."&k
=".$data['kend']." target=_blank>monitor</a></td>";
    $i++;
}
echo "</table>";
echo "<br><br>";

$querypage    = "SELECT COUNT(*) AS jumData FROM
kendaraan";
$hasilpage    = mysql_query($querypage);
$datapage     = mysql_fetch_array($hasilpage);

$jumData = $datapage['jumData'];

// menentukan jumlah halaman yang muncul berdasarkan jumlah
semua data

$jumPage = ceil($jumData/$dataPerPage);

// menampilkan link previous

if      ($noPage    >    1)      echo
            "<a
href=\"$_SERVER['PHP_SELF'].\"?paging=\".($noPage-
1).\">&lt;&lt; Prev</a>";

// memunculkan nomor halaman dan linknya

for($page = 1; $page <= $jumPage; $page++)
{
    if (((($page >= $noPage - 3) && ($page <= $noPage + 3)) ||
($page == 1) || ($page == $jumPage))
    {
        if (($showPage == 1) && ($page != 2)) echo "...";
    }
}

```

```

        if (($showPage != ($jumPage - 1)) && ($page == $jumPage))
echo "...";
        if ($page == $noPage) echo " <b>".$page."</b> ";
        else echo " <a
href='$_SERVER['PHP_SELF']. "?paging=".$page.">".$page."</a>
";
        $showPage = $page;
    }
}

// menampilkan link next

    if ($noPage < $jumPage) echo " <a
href='$_SERVER['PHP_SELF']. "?paging=".( $noPage+1).">Next
&gt;&gt;</a>";

    ?>

</br>
</br>

```

### Proses Tap-in

```

<form action="tap_in.php" method="post" target=_blank>
kartu :
<select name="id">
    <?php
        $koneksi = mysqli_connect("localhost","root","","simulasi");

        $query = "select * from tiket";
        $hasil = mysqli_query($koneksi,$query);
        while($data=mysqli_fetch_array($hasil)){
            echo "<option value=$data[id]>$data[id]</option>";
        }
    ?>
</select>
<br>
    <script type="text/javascript" src="jquery-1.4.2.min.js"></script>
Kendaraan: <select name="kendaraan" id="kendaraan">

```



```

<option value="">- Pilih Kendaraan -</option>

<!-- looping data provinsi -->
<?php
$type="select * from kendaraan";
$type1=mysql_query($type);
while($data=mysql_fetch_array($type1)){

    ?>
        <option value="<?php echo $data["kend"] ?>"><?php echo
        $data["kend"] ?></option>

    <?php
    }
    ?>
</select>

<br>
halte naik : <select name="halte" id="halte">
    <!-- hasil data dari cari_kota.php akan ditampilkan disini -->
</select>

<script>

    $("#kendaraan").change(function(){

        // variabel dari nilai combo box provinsi
        var type = $("#kendaraan").val();

        // tampilkan image load

        // mengirim dan mengambil data
        $.ajax({
            type: "POST",
            dataType: "html",
            url: "cari_halte.php",
            data: "type="+type,
            success: function(msg){

```

```

        // jika tidak ada data
        if(msg == ""){
            alert('Tidak ada data daftar halte');
        }

        // jika dapat mengambil data,, tampilkan di combo box kota
        else{
            $("#halte").html(msg);
        }

        // hilangkan image load
        //$("#imgLoad").hide();
    }
    });
});
</script></br>
<input type="submit" value="Submit">
</form>
</br>
</br>
Proses Tap-out

<form action="tap_out.php" method="post" target=_blank>
    <script type="text/javascript" src="jquery-1.4.2.min.js"></script>
    Kartu: <select name="kartu" id="kartu">
        <option value="">- Pilih Kartu -</option>

        <!-- looping data provinsi -->
        <?php
            $kartu="select * from rekap where userin=1";
            $kartu1=mysql_query($kartu);
            while($data=mysql_fetch_array($kartu1)){

                ?>
                <option value="<?php echo $data["kartu"] ?>"><?php echo
                $data["kartu"] ?></option>

                <?php
            }

```

```

?>
</select>

<br>
halte turun : <select name="halte" id="halte1">
  <!-- hasil data dari cari_kota.php akan ditampilkan disini -->
</select>

<script>

$("#kartu").change(function(){

    // variabel dari nilai combo box provinsi
    var kartu = $("#kartu").val();

    // tampilkan image load

    // mengirim dan mengambil data
    $.ajax({
        type: "POST",
        dataType: "html",
        url: "cari_halte2.php",
        data: "kartu="+kartu,
        success: function(msg){

            // jika tidak ada data
            if(msg == ""){
                alert("Tidak ada data daftar halte");
            }

            // jika dapat mengambil data,, tampilkan di combo box kota
            else{
                $("#halte1").html(msg);
            }

            // hilangkan image load
            //$("#imgLoad").hide();
        }
    });
});

```

```

    });
</script></br>
    <input type="submit" value="Submit">
</form>
</body>
</html>

```

### **-Halaman ‘Daftar Tiket’**

```

<!DOCTYPE html>
<html>
<head>
    <title>Simulasi e-ticketing</title>
</head>
<body>
<table>
<tr>
<td>
<form action="index.php">
    <input type="submit" value="Halaman Utama">
</form></td>
<td>
<form action="tiket.php">
    <input type="submit" value="Daftar Tiket">
</form></td>
<td>
<form action="live.php">
    <input type="submit" value="Update Live">
</form></td>
<td>
<form action="rekap2.php">
    <input type="submit" value="Rekap Akhir">
</form></td></tr>
</table>
<h2>Daftar Tiket</h2>
<?php
include "db.php";
error_reporting(0);
// jumlah data yang akan ditampilkan per halaman

```

```

$dataPerPage = 10;

// apabila $_GET['page'] sudah didefinisikan, gunakan nomor halaman
tersebut,
// sedangkan apabila belum, nomor halamannya 1.

if(isset($_GET['paging']))
{
    $noPage = $_GET['paging'];
}
else $noPage = 1;

// perhitungan offset

$offset = ($noPage - 1) * $dataPerPage;

$query=mysql_query("select*from      tiket      LIMIT      $offset,
$dataPerPage");
echo "
<table border=1><tr>
<th width=50 >No.</th>
<th width=100>ID RFID</th>
<th width=100>Saldo</th>
<th width=100>5000</th>
<th width=100>10000</th>
</tr>";
$i=1;
while ($data=mysql_fetch_array($query)) {
echo "<tr>";
echo "<td>".$i."</td>";
echo "<td>".$data['id']."</td>";
echo "<td>".$data['saldo']."</td>";
echo
                                "<td><a
href=tambahsaldo.php?id=".$data['id']."&jum=5>Tambah</a></td>";
echo
                                "<td><a
href=tambahsaldo.php?id=".$data['id']."&jum=10>Tambah</a></td>";
";
    $i++;
}

```

```

}
echo "</table>";
echo "<br><br>";

$querypage = "SELECT COUNT(*) AS jumData FROM tiket";
$hasilpage = mysql_query($querypage);
$dapage = mysql_fetch_array($hasilpage);

$jumData = $dapage['jumData'];

// menentukan jumlah halaman yang muncul berdasarkan jumlah
semua data

$jumPage = ceil($jumData/$dataPerPage);

// menampilkan link previous

if ($noPage > 1) echo "<a
href=\"$_SERVER['PHP_SELF'].\"?paging=\".$noPage-
1).\">&lt; Prev</a>";

// memunculkan nomor halaman dan linknya

for($page = 1; $page <= $jumPage; $page++)
{
    if (($page >= $noPage - 3) && ($page <= $noPage + 3)) ||
($page == 1) || ($page == $jumPage))
    {
        if (($showPage == 1) && ($page != 2)) echo "...";
        if (($showPage != ($jumPage - 1)) && ($page == $jumPage))
echo "...";
        if ($page == $noPage) echo "<b>.$page.</b> ";
        else echo "<a
href=\"$_SERVER['PHP_SELF'].\"?paging=\".$page.\">.$page.</a>
";
        $showPage = $page;
    }
}

```

```
// menampilkan link next

if ($noPage < $jumPage) echo "<a
href=\"$_SERVER['PHP_SELF'].\"?paging=\".($noPage+1).\">Next
&gt;&gt;</a>";

?>

</body>
</html>
```

### **-Halaman ‘Rekap Akhir’**

```
<!DOCTYPE html>
<html>
<head>
  <title>Simulasi e-ticketing</title>
</head>
<body>
  <table>
    <tr>
      <td>
        <form action="index.php">
          <input type="submit" value="Halaman Utama">
        </form></td>
      <td>
        <form action="tiket.php">
          <input type="submit" value="Daftar Tiket">
        </form></td>
      <td>
        <form action="live.php">
          <input type="submit" value="Update Live">
        </form></td>
      <td>
        <form action="rekap2.php">
          <input type="submit" value="Rekap Akhir">
        </form></td></tr>
    </table>
    <h2>Rekapitulasi E-Ticketing</h2>
```

```

<?php
include "db.php";
error_reporting(0);
// jumlah data yang akan ditampilkan per halaman

$dataPerPage = 10;

// apabila $_GET['page'] sudah didefinisikan, gunakan nomor halaman
tersebut,
// sedangkan apabila belum, nomor halamannya 1.

if(isset($_GET['paging']))
{
    $noPage = $_GET['paging'];
}
else $noPage = 1;

// perhitungan offset

$offset = ($noPage - 1) * $dataPerPage;

$query=mysql_query("select*from    rekap2    LIMIT    $offset,
$dataPerPage");
echo "
<table border=1><tr>
<th width=50 >No.</th>
<th width=100>Kendaraan</th>
<th width=100>kartu</th>
<th width=100>Latitude</th>
<th width=100>Longitude</th>
<th width=100>User masuk</th>
<th width=100>User keluar</th>
<th width=100>Waktu</th>
</tr>";
$i=1;
while ($data=mysql_fetch_array($query)) {
echo "<tr>";
echo "<td>".$i."</td>";
echo "<td>".$data['kend']."</td>";

```



```

echo "<td>".$data['kartu']. "</td>";
echo "<td>".$data['latitude']. "</td>";
echo "<td>".$data['longitude']. "</td>";
echo "<td align=center>".$data['userin']. "</td>";
echo "<td align=center>".$data['userout']. "</td>";
echo "<td width=175 align=center>".$data['waktu']. "</td>";
$i++;
}
echo "</table>";
echo "<br><br>";

$querypage = "SELECT COUNT(*) AS jumData FROM rekap2";
$hasilpage = mysql_query($querypage);
$datapage = mysql_fetch_array($hasilpage);

$jumData = $datapage['jumData'];

// menentukan jumlah halaman yang muncul berdasarkan jumlah
semua data

$jumPage = ceil($jumData/$dataPerPage);

// menampilkan link previous

if ($noPage > 1) echo "<a
href='".$_SERVER['PHP_SELF']."'?paging=".$noPage-
1.">&lt;&lt; Prev</a>";

// memunculkan nomor halaman dan linknya

for($page = 1; $page <= $jumPage; $page++)
{
    if ((($page >= $noPage - 3) && ($page <= $noPage + 3)) ||
($page == 1) || ($page == $jumPage))
    {
        if (($showPage == 1) && ($page != 2)) echo "...";
        if (($showPage != ($jumPage - 1)) && ($page == $jumPage))
echo "...";
        if ($page == $noPage) echo " <b>".$page."</b> ";
    }
}

```

```

        else echo "
href='$_SERVER['PHP_SELF'].'?paging=".$page.">".$page."</a>
";
        $showPage = $page;
    }
}

// menampilkan link next

if ($noPage < $jumPage) echo "<a
href='$_SERVER['PHP_SELF'].'?paging=".( $noPage+1 )."'>Next
&gt;&gt;</a>";

?>

</br>
</br>
</br>
<a href="truncate2.php">kosong kan isi tabel</a>

</body>
</html>

```

## DAFTAR PUSTAKA

- [1] B. Feng, A. Lakshminarayanan dan D. Robert, “Design of Portable Mobile Devices Based E-Payment System and E-Ticketing System with Digital Signature,” Kent Ridge Digital Labs, Singapore, 2001.
- [2] S. B. William, A. Lumenta dan X. Najoran, “Analisis Kualitas Layanan Jaringan Internet (Studi Kasus PT. Kawanua Internetindo Manado),” UNSRAT, Manado, 2014.
- [3] K. Pemerintah, “Surabaya Mass Rapid Transportation,” Surabaya, 2013.
- [4] M. Mezghani, “Study on electronic ticketing in public transport,” European Metropolitan Transport Authorities, Eropa, 2008.
- [5] M. B. Sardar, O. Mazliza dan R. K. Atta, “A Performance Comparison of Open Source Network Simulators for Wireless Network,” IEEE International Conference on Control System, Computing and Engineering, Malaysia, 2012.
- [6] IEEE 802 Series, “Local Area Network and Metropolitan Area Network,” IEEE, 2009.
- [7] ITU-T Series G, “Transmission system and media, digital system and network,” 2003.
- [8] A. Khadir, Tutorial Network Simulator 2, 2011.
- [9] Assisten lab CnC It Telkom, “Modul Benginpro 2014,” CnC Laboratory, Bandung, 2014.
- [10] Tiphon, “Telecommunications and Internet Protocol Harmonization Over Network (TIPHON) General aspects of Quality of Service (QoS),” DTR/TIPHON-05006 (cb0010cs.pdf), 1999.

## RIWAYAT HIDUP



Penulis bernama lengkap Prasetyo Yuliantoro. Lahir di Yogyakarta, 20 Juli 1992, merupakan anak kedua dari dua bersaudara. Dibesarkan di kota Yogyakarta, penulis memulai pendidikan formal di SDN Kabregan pada tahun 1998, kemudian melanjutkan pendidikan di SMPN 1 Piyungan pada tahun 2004, dan pada jenjang berikutnya melanjutkan pendidikan di SMAN 1 Kalasan hingga lulus pada tahun 2010. Setamat dari SMA, penulis meneruskan pendidikan di D3 Jurusan Teknik Telekomunikasi, Universitas Telkom Bandung pada tahun 2010 sampai 2013. Pada tahun yang sama melanjutkan tingkat S1 di Institut Teknologi Sepuluh November dengan jurusan Teknik Elektro dengan Prodi Telekomunikasi. Penulis dapat dihubungi melalui alamat email : [prasetyo.saka@gmail.com](mailto:prasetyo.saka@gmail.com)